

Received 11 September 2022, accepted 20 September 2022, date of publication 26 September 2022, date of current version 4 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3209824

RESEARCH ARTICLE

Performance Enhancement of In-Vehicle 10BASE-T1S Ethernet Using Node Prioritization and Packet Segmentation

JIHYEON MIN AND YOUNGIL PARK¹

Department of Electronic Engineering, Kookmin University, Seoul 02707, South Korea

Corresponding author: Youngil Park (ypark@kookmin.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2021R1F1A1048227.

ABSTRACT Following the appearance of electrical vehicles and autonomous driving, a new in-vehicle network architecture is required that should be able to process substantial sensor data and communicate with other vehicles or infrastructure. Ethernet is considered a promising technology for replacing existing communication networks due to its stability and large bandwidth. Among various types of Ethernet, 10BASE-T1S can play a significant role in connecting multiple nodes in a bus structure at each zone of the zone-based network architecture. Although its latency is reduced using the physical layer collision avoidance (PLCA) algorithm, it is not small enough to be adopted in safety and powertrain domains, which require a very small delay of less than a few hundred microseconds. Therefore, this study uses node prioritization and packet segmentation to overcome the limitations of the existing PLCA algorithm. The former changes the transmission sequence of nodes while the latter reduces the waiting time for a packet. This paper suggests the algorithms of these schemes and analyzes the performance.

INDEX TERMS In-vehicle network, automotive Ethernet, 10BASE-T1S Ethernet, physical layer collision avoidance (PLCA), prioritization, segmentation.

I. INTRODUCTION

With the evolution of vehicular technologies, more complex applications such as advanced driver assistance systems, autonomous driving, enhanced safety, and infotainment, which require wide bandwidth, have been adopted in vehicles [1]. Automotive Ethernet has gained more attention to meet this requirement because it can provide a wide bandwidth and fault-tolerant connectivity among various subsystems at a low cost [2]. Furthermore, it has already been verified in the industry. Among the diverse types of automotive Ethernet, 100BASE-T1 satisfies the requirements for bandwidth, delay, synchronization and network management of vehicular networks. It is good for implementing a network in a tree topology, which requires a lot of switches to

connect many electronic control unit (ECU) nodes as shown in Fig. 1(a) [3].

However, many existing ECUs are connected in a bus topology such as controller area network (CAN) and local interconnect network (LIN). The bus network is suitable for connecting many ECUs in the vehicle because it reduces the cable length compared to the point-to-point structure. The 10BASE-T1S that operates at 10 Mbps in the bus network can replace existing CAN and LIN [4]. It has other advantages such as light cable weight, simple equalization; it does not need forward error correction, echo cancellation and hybrid circuit [5]. Therefore, it will be used efficiently in a zone-based in-vehicle network as shown in Fig. 1(b), which can reduce the number of ECUs and total cable length. Zonal architecture divides the vehicle into several zones, where each zone gateway communicates with ECU nodes inside its zone. Since many existing ECUs are connected in bus topology, 10BASE-T1S can be used for this purpose.

The associate editor coordinating the review of this manuscript and approving it for publication was Leandros Maglaras¹.

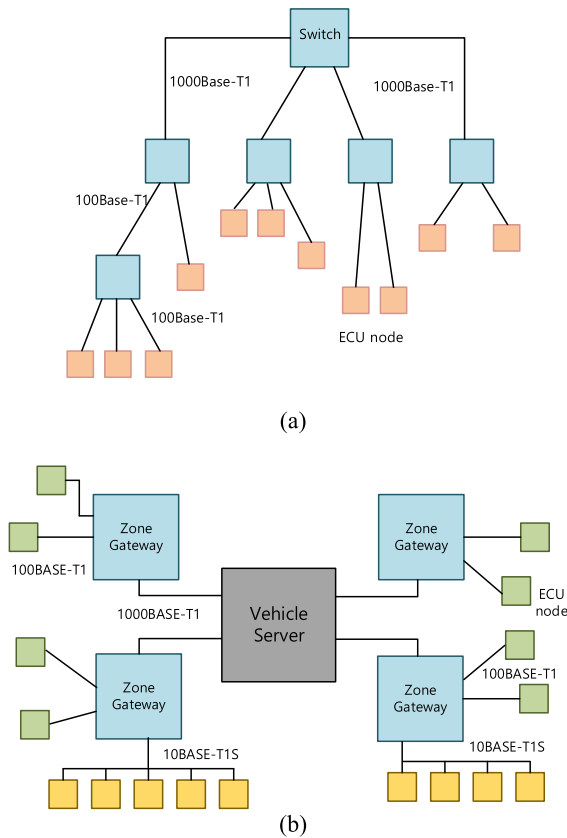


FIGURE 1. In-vehicle network architectures (a) tree topology network (b) zone-based network.

The gathered data from ECUs inside a zone are transmitted to other zones through the backbone network, where high-speed automotive Ethernet with greater than 1 Gbps data rate can be used. Therefore, it requires less cable than domain-based architecture [6]. A CPU at the center of the zones, called a vehicle server, processes the sensor data generated from all zones. Security should be ensured in this architecture since data from different domains are exchanged at the vehicle server [7].

TABLE 1. Properties of in-vehicle networks [8].

Parameters	CAN(-FD)	LIN	FlexRay
Topology	Bus	Bus	Bus/Star
Data rate	Max. 1 Mbps (-FD : 8 Mbps)	Max. 20 kbps	Max. 10 Mbps
Range	40 m	40 m	10 m
Max Data	8 Bytes (-FD : 64 Bytes)	8 Bytes	254 Bytes
Usage	Powertrain, Control, Diagnosis	Body, Connector	X-by-wire, safety

Properties of the traditional in-vehicle networks are explained in Table 1 [8] for the comparison with that of 10BASE-T1S. In particular, CAN could be used in the transmission of delay-sensitive data by assigning a small ID to a high-priority data [8], [9]. When a collision happens

among frames from different nodes, IDs of them are compared; only the frame with the smallest ID is allowed to continue its transmission while other frames stop transmission. In this way, urgent messages are transmitted with low latency without interruption from other nodes in the same bus.

If 10BASE-T1S is to replace or cooperate with CAN, its latency should be very small. Latency requirements for in-vehicle networks are dependent on domains. Backbone communication generally requires an end-to-end latency of less than 10 msec [10], [11]. In contrast, the control loop signal generated from an existing CAN device requires a latency of less than 100 μ sec. The additional delay should be considered if the data pass through the gateway. Fig. 2 explains the properties and latency requirements for each domain. According to the figure, safety and engine/powertrain domains have very tight latency requirements, whereas comfort and Human-Machine-Interface (HMI) domains have a relatively generous limit. Concerning the periodic messages usually found in powertrain domains, the maximum allowed delay is about 10 % of the period [12]. For instance, when the period of a control loop signal for controlling a motor or an actuator is 1 msec, the delay limit is 0.1 msec.

The 10BASE-T1S, which uses the same bus topology as the existing CAN, utilizes the physical layer collision avoidance (PLCA) function, reducing the delay by preventing collisions in the physical layer. However, it is difficult to satisfy all latency requirements because each node must wait for its turn according to the round-robin-based protocol. Even urgent nodes should wait until all preceding nodes complete their transmissions. Assuming maximum Ethernet packet length (i.e., 1,530 bytes) with 8 nodes and 10 Mbps data rate the delay reaches up to 9.8 msec, which exceeds the delay limit of engine/powertrain and safety electronics as shown in Fig. 2. Therefore, we reduce the waiting time by assigning higher priority to these delay-sensitive nodes. Even in this case, the waiting time from the packet transmitted just before the priority packet is inevitable. One way to decrease it is to reduce the packet size. For this purpose, we propose a packet segmentation.

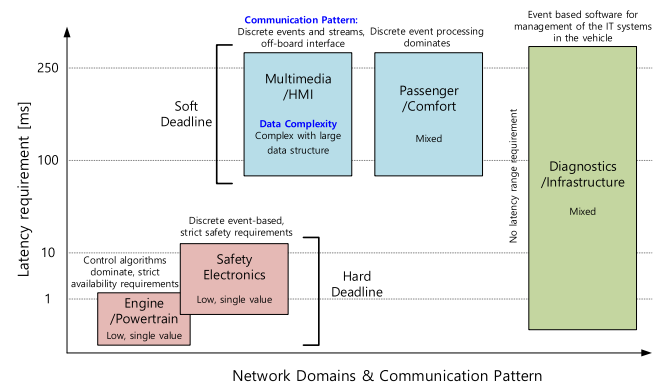


FIGURE 2. Latency requirements for each domain in vehicular networks.

There are studies to achieve low latency in other layers of vehicular communications. A network layer approach is

essential to secure the performance in inter-vehicle network. One of the examples is ARTNet (Ai-Based Resource Allocation and Task Offloading in a Reconfigurable Internet of Vehicular Networks), which maximizes resource utilization and optimizes the distribution of traffic [13]. Although it may be helpful to avoid large delay in the Internet of Vehicular (IoV) infrastructure, it cannot prevent the network delay inside an individual vehicle. One of the MAC layer approaches is the time-sensitive network (TSN), which uses a time-aware scheduler and virtual local area network (VLAN) tag to perform prioritization and fragmentation functions. The IEEE 802.1Q TSN standards provide synchronous or asynchronous real-time traffic classes to meet real-time and robust requirements [14]. However, it is challenging to use the TSN in the multidrop mode of 10BASE-T1S [15]. The prioritization of the TSN cannot be used at the physical layer (PHY) because the VLAN tag value of the packet must be checked at the media access control (MAC) layer. In addition, the TSN requires multiple queues to connect to different output ports, which are appropriate for a tree architecture. There also has been researches to reduce the latency in the physical layer. A priority-based PLCA research was conducted to provide transmission opportunities by subdividing the cycle according to the priority of nodes and their messages [16]. If a high-priority message occurs in a sequence of low priority messages, the cycle is restarted to transmit the high-priority message first. However, data with low priority, in this case, can experience too much delay due to the extended cycle length. A study has used the concept of priority in CAN-over-PLCA Ethernet [17]. In this case, however, only one priority node exists; furthermore, it is insufficient to meet the latency requirements of all nodes because packet segmentation is not used.

This paper is organized as follows. Section II explains the concept of the existing PLCA and the application of priority in PLCA. In addition, the zonal architecture to which the proposed architecture can be applied is described. Section III presents the packet segmentation and a state diagram depicting its implementation in the physical layer. Section IV analyzes the performance of the proposed algorithm, and Section V concludes the study.

II. PRIORITY-BASED PHYSICAL LAYER COLLISION AVOIDANCE

This section explains the PLCA function of the 10BASE-T1S Ethernet and its delay. Then, its enhanced version, priority-based PLCA, is proposed. In this case, nodes with high priority transmit data prior to other nodes. The performance of the proposed method is compared with that of the existing PLCA.

A. ZONE-BASED ETHERNET ARCHITECTURE

The in-vehicle network will likely evolve from the existing domain-based structure to a zone-based structure because the latter can make the Ethernet-based network integration and help the over-the-air (OTA) software update easily. It will

simplify the network structure in terms of the number of ECUs and cable length, although it requires a powerful server [6]. Fig. 3 illustrates the zone-based Ethernet where the 10BASE-T1S bus is integrated.

The high-speed Ethernet backbone connects each zone to the vehicle server [7]. Ethernet backbone can provide deterministic, high bandwidth and fault-tolerant connectivity [18]. The zone gateway, positioned at the center of each zone, is connected to many ECUs inside the zone through the automotive Ethernet in a star or bus architecture as illustrated in Fig. 3. Each node in the 10BASE-T1S Ethernet can operate as an independent PLCA node or gateway for the CAN or CAN-FD (flexible data rate) bus. If it is available at very low cost, then it will replace CAN or CAN-FD; but at the present time, it is likely to coexist with them via CAN-to-Ethernet conversion at the gateway [19].

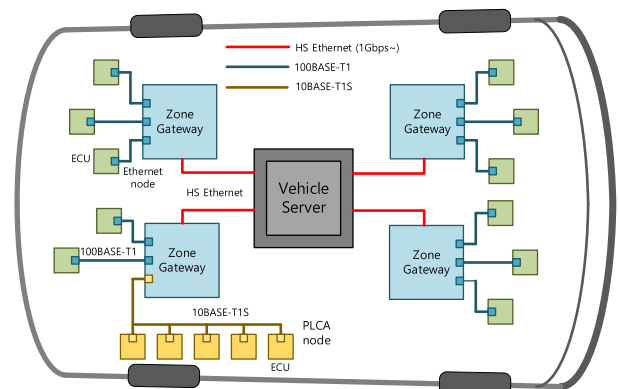


FIGURE 3. Zone-based architecture for the automotive Ethernet.

Fig. 4 indicates how the CAN-FD frame can be mapped to the PLCA Ethernet packet. The EtherType in the Ethernet packet includes information on the CAN-FD frame, whereas the payload includes the data and overhead of the CAN-FD. The CAN protocol guarantees low latency to delay-sensitive data through the arbitration process using the CAN ID. However, PLCA does not have this function, and low latency cannot be achieved for time-critical data. This paper introduces node prioritization and packet segmentation in PLCA to satisfy the delay requirement. This concept is not limited to the zone-based structure, but more applications can be found in this architecture.

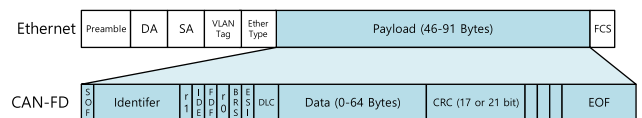


FIGURE 4. Frame mapping of CAN-FD to 10BASE-T1S Ethernet.

B. 10BASE-T1S ETHERNET AND PHYSICAL LAYER COLLISION AVOIDANCE

The 10BASE-T1S is an Ethernet standard developed as a part of IEEE 802.3cg. Its PHY supports half-duplex

communication that can interconnect up to eight nodes in a trunk that reaches up to 25 m. In this standard, PLCA is a function that prevents the packet collision in a physical layer by assigning a transmission opportunity (TO) to each node in a round-robin fashion, that only the node with the TO is allowed to transmit a single packet. It cooperates with carrier sense multiple access with collision detection (CSMA/CD) protocol in half-duplex shared-medium networks and provides enhanced bandwidth and improved access latency under heavily loaded traffic conditions. The PLCA reconciliation sublayer aligns data from the MAC with the TO in the PHY and maps the PHY signals to physical signaling sublayer (PLS) primitives towards the MAC [4]. The 10BASE-T1S Ethernet bus comprising $N + 1$ nodes is illustrated in Fig. 5. Each node has a unique node ID, and the node with ID 0 is called a primary node in this paper, whereas all other nodes are secondary nodes.

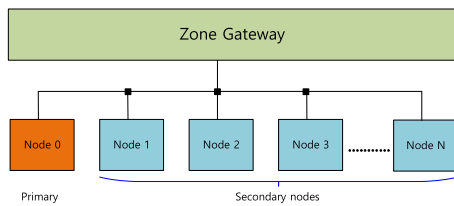


FIGURE 5. Bus architecture of 10BASE-T1S Ethernet.

Fig. 6 illustrates the node transmission sequence when the PLCA function is working. In the beginning, the primary node sends a BEACON signal to start a cycle and initialize the TO. When each node gains a TO, it transmits a COMMIT signal and an Ethernet packet. If the node with the TO has no packet to send, it relinquishes the opportunity and transmits nothing for a certain period; this process is called SILENCE. As each node acts, other nodes increase the TO number and wait until their ID reaches it. When a cycle is completed up to the last node, the primary node sends the BEACON signal again to reinitialize the cycle.

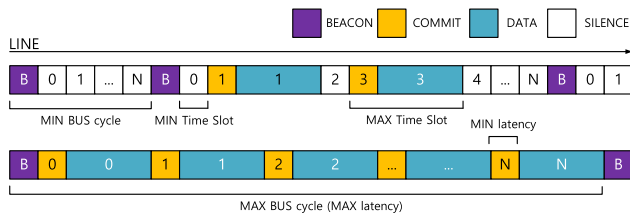


FIGURE 6. Transmission sequence of each node with PLCA.

The minimum cycle in Fig. 6 is found when all nodes yield their TOs, whereas the maximum cycle occurs when all nodes transmit packets at the maximum length. If there are $N + 1$ nodes with IDs from 0 to N , the minimum and maximum lengths of the PLCA cycle are stated as in Eqs. (1) and (2). In these equations, R_b denotes the data rate, L_{beacon} is the BEACON timer length, L_{to} represents the timeout timer length counting the TOs to yield, L_{commit} denotes the

COMMIT signal length, and $L_{data,max}$ refers to the maximum Ethernet packet (i.e. 1,530 bytes, including the header and frame check sequence). When the data rate is 10 Mbps, and N is 7, the minimum cycle length corresponds to $27.6 \mu\text{sec}$. Under the same conditions, the maximum length becomes 9.8 msec, too large for delay-sensitive data. The existing PLCA scheme cannot meet the hard latency requirements in Fig. 2; thus, this study introduces a priority-based PLCA algorithm.

$$T_{cycle,min} = \frac{1}{R_b} \{L_{beacon} + (N + 1) \cdot L_{to}\} \quad (1)$$

$$T_{cycle,max} = \frac{1}{R_b} \{L_{beacon} + (N + 1) (L_{commit} + L_{data,max})\} \quad (2)$$

C. PRIORITY ASSIGNMENT ALGORITHM FOR THE PHYSICAL LAYER COLLISION AVOIDANCE

The delay that occurs from the PLCA protocol may exceed the latency requirement of some ECU node. In order to overcome this problem, nodes in a 10BASE-T1S bus are classified into regular or priority nodes. Delay-sensitive nodes such as engine/powertrains or safety electronics can be categorized as priority nodes; a unique priority number is stored at each node.

The priority assignment in this study utilizes the interpacket gap (IPG) to indicate the priority request information. The IPG is the period of at least 12 bytes after each Ethernet packet. The transmitting node is prohibited from sending data during this period. This IPG is divided into multiple request slots and a control message slot as depicted in Fig. 7. The formers are used to request priority transmissions and the latter to send PLCA control messages. Each priority node can send transmission request messages in its assigned slot when it has data to send. Then, it sends a COMMIT message in the control section and the Ethernet packet. In this way, the priority node can send its payload prior to other nodes. In this process, once the higher-priority section is used, the lower-priority sections should be left empty to avoid transmission conflict. If there is no priority node to send a message, the node in the original round-robin sequence takes the TO. In this paper, section IV analyzes the performance by assuming two priority nodes, which usually transmit short packets from the CAN or CAN-FD bus.

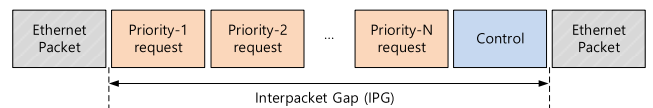


FIGURE 7. Request for priority data transmission in the interpacket gap.

This study assumes that the bus comprises eight nodes, where two nodes are assigned priority levels. The node with ID 0 (or Node 0) is the primary node, and Nodes 3 and 5 are assumed to be priority nodes. Node 3 has higher priority and can send a request message in the Priority-1 slot, whereas

Node 5 can use the Priority-2 slot in Fig. 7. The operation of the PLCA in the absence of a priority packet is depicted in Fig. 8(a), where each node has a TO in sequence. If it has data, it sends the COMMIT message first, followed by the Ethernet packet. If it has no data, the SILENCE period occupies the slot. Once the Ethernet packet is sent, the IPG period follows, where the priority request and control messages are included. After the transmission of Node 2 in Fig. 8(a), Node 3 surrenders its opportunity during the IPG period, and Node 4 takes the TO and sends data.

If a priority node has data to transmit, it sends a request message to have the TO during the coming IPG period. Fig. 8(b) illustrates the operation of PLCA when data are generated in both priority nodes, Nodes 5 and 3, in sequence, during transmission of Node 1. As Node 3 has higher priority, it is allowed to request a priority message first and takes the TO. Then, Node 5 takes its turn to transmit. Node 2 can use the bus according to the round-robin sequence if the priority nodes finish their transmissions.

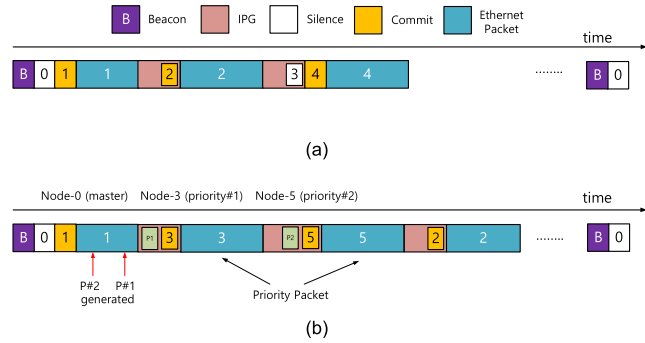


FIGURE 8. Operation of physical layer collision avoidance (a) without priority data (b) with priority data.

The latency of the proposed scheme can be calculated as follows. A higher-priority node is referred to as P#1, whereas a lower-priority node is P#2. It is assumed that the generated data are stored in a buffer and transmitted when a TO is given. However, if the data are generated during the transmission at their own node, the transmission should wait until the next TO. As depicted in Fig. 9, the maximum latency of the P#1 occurs when the node that transmitted data just before sent a full-sized packet, whereas the maximum latency of P#2 occurs when both P#1 and the previous node transmit maximum-sized packets.

Therefore, the maximum latencies of P#1 and P#2 are written as Eqs. (3) and (4), respectively.

$$T_{delay,max,p\#1} = \frac{1}{R_b} \{ (PLCA \text{ node no.} - 2) \times L_{to} + L_{beacon} + L_{commit} + L_{data,max} + IPG \} \quad (3)$$

$$T_{delay,max,p\#2} = \frac{1}{R_b} \{ (PLCA \text{ node no.} - 2) \times L_{to} + L_{beacon} + L_{commit} + (L_{data,max} + IPG) + (L_{priority,max} + IPG) \}, \quad (4)$$

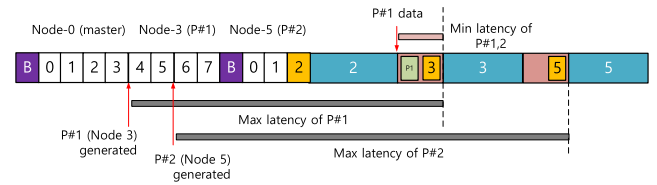


FIGURE 9. Maximum and minimum latencies of priority-based physical layer collision avoidance.

where $PLCA \text{ node no}$ denotes the number of nodes in the PLCA bus, and $L_{priority,max}$ denotes the maximum-sized packet of the priority nodes. If R_b is 10 Mbps, $PLCA \text{ node no}$ is 8 bits, L_{to} is 32 bits, L_{beacon} is 20 bits, L_{commit} is 5 bits, $L_{data,max}$ is 1,530 bytes, $L_{priority,max}$ is 121 bytes and IPG is 12 bytes, then, the maximum latencies of P#1 and P#2 are 1.3 and 1.36 msec, respectively. $L_{priority,max}$ is assumed to be 121 bytes because the maximum size of the CAN-FD frames is 91 bytes, and the header and frame check sequence in its Ethernet mapping correspond to 30 bytes, as presented in Fig. 4 [20]. Compared to the maximum cycle length of 9.8 msec in conventional PLCA, the latency of the proposed priority node is reduced to 1.36 msec, almost 1/7 of its original value. Minimum latency occurs when data are generated just before their TO, as indicated in Fig. 9. It is calculated as in Eq. (5), the same as in the existing PLCA.

$$T_{delay,min} = L_{commit} \div R_b \quad (5)$$

III. SEGMENTATION OF AN ETHERNET PACKET

The previous section states that node prioritization can reduce the latency of urgent nodes. However, the reduced latency is not small enough to satisfy the requirements of ECUs in all domains shown in Fig. 2. Therefore, the concept of packet segmentation is proposed in this section to further reduce the latency.

A. PRINCIPLES OF SEGMENTATION

By introducing priority nodes in 10BASE-T1S Ethernet, the maximum latency can be reduced to 1.36 msec, enabling the network to support more services listed in Fig. 2. However, the reduced latency still does not satisfy the requirements of engine/powertrain-related ECUs. The limit in the latency reduction is attributed to the Ethernet packet size. As the 10BASE-T1S Ethernet uses the CSMA/CD protocol, even a high-priority node must wait until the end of the current data transmission. Therefore, the packet length affects the delay, and we propose packet segmentation to further reduce the delay. Unlike priority data, which usually have a short length, a normal Ethernet packet has a long payload of up to 1,530 bytes. According to the proposed segmentation, a packet longer than a predefined size is divided into multiple segments. Then, each segment is transmitted in one cycle. Therefore, the segment size should be determined by considering the required latency. Packet segmentation has been used in Gigabit Passive Optical Network (GPON), TTEthernet

and TSN, where packets are fragmented in the MAC layer. However, in this paper, segmentation is combined with PLCA in the reconciliation sublayer.

Fig. 10 depicts the control message in the IPG to use prioritization and segmentation in the PLCA-based Ethernet. Each node transmits the node ID and segment information after transmitting a COMMIT message. The node ID is required to differentiate segments from multiple nodes, and the segment information is needed to indicate whether the packet is segmented and is the last segment. The node ID and segment information are used to reassemble multiple segments in PLCA. If the length of an Ethernet packet exceeds the predefined size, this packet is divided into multiple segments, as displayed in Fig. 11. Among them, the first segment includes the overhead and front parts of the packet. The following segments comprise preambles and the following parts of the packet, whereas the last one has a frame check sequence part for error detection and correction. Each segment is transmitted once in a cycle. Therefore, multiple cycles are required to send a long Ethernet packet.

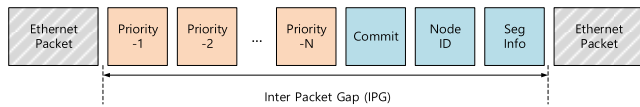


FIGURE 10. Assignment of priority and segmentation in the interpacket gap.

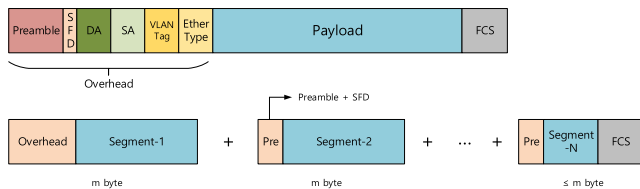


FIGURE 11. Segmentation of Ethernet packet in physical layer collision avoidance.

When a node receives data, it checks the segment information to determine whether the data are segmented and whether the packet is the last one. If the received packet is not segmented, it is immediately delivered to the upper MAC layer. However, if it is segmented, when the received packet is the last segment, all stored segments are reassembled and delivered to the MAC layer. The latency of this transmission can be calculated using Eqs. (3) and (4), but it should be added multiple times for the latency of multiple segments. For instance, if the maximum payload length in a segment is 200 bytes, the maximum delay is 0.215 msec for the first priority node, P#1, and 0.322 msec for the second priority node, P#2. Otherwise, if the maximum payload size is 100 bytes, the maximum latency is 0.135 msec for P#1 and 0.242 msec for P#2. The delay of the priority node is reduced more as the segment size becomes smaller. However, it is noted that the segmentation causes increased ratio of overhead in the Ethernet frame from regular nodes, which comprises preamble, start frame delimiter (SFD) and IPG. As the length of it

amounts to 20 bytes while the size of the Ethernet packet is reduced due to the segmentation, it makes the regular nodes experience more latency. Therefore, the size of a segment should be chosen considering latency boundaries of both the priority packets and the regular packets. The effect of the segment sizes is analyzed in section IV.

B. ALGORITHM FOR PACKET SEGMENTATION

This section explains the implementation algorithm for the proposed packet segmentation in more detail. Fig. 12 presents the operation of the reconciliation sublayer, where function blocks of PLCA are placed [4]. In addition to the existing PLCA functions, two more blocks, the SEGMENT_CTRL and REASSEMBLY_CTRL blocks indicated in color blocks, are added to implement the segmentation. The SEGMENT_CTRL block is responsible for the packet segmentation before transmission. If the packet from the PLS is longer than the predefined size, it is segmented and stored in a buffer. Once it takes the TO, it sends a tx_cmd signal, such as COMMIT, node ID and segment information, followed by a segment via the TXD <3:0> bus. This block sends the COL signal to the PLS until the buffer is empty so that the MAC does not transmit a new packet. The REASSEMBLY_CTRL block saves the received segments and reassembles them into a complete packet. For this purpose, it contains multiple buffers to store different segments according to their node IDs. Checking the segment information messages determines whether to reassemble the segments in the buffer and forward them to the MAC layer.

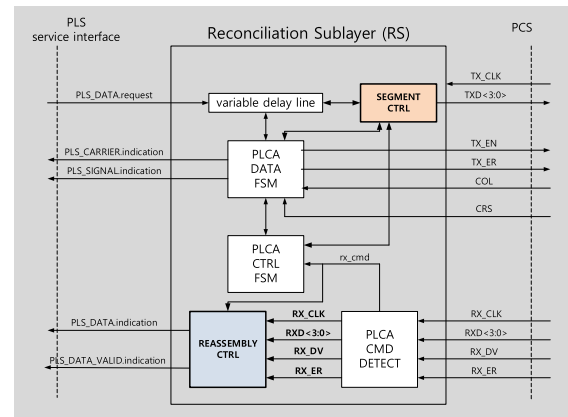


FIGURE 12. Operation of the reconciliation sublayer for segmentation.

Fig. 13 illustrates the state diagram of the PLCA control, which is executed at the PLCA_CTRL_FSM block in Fig. 12. The CHECK_PRIORITY block is connected to the transmission and reception blocks, and the priority timer works during the IPG period. If there is a priority packet to transmit at this node, it waits for the IPG interval assigned in the SEND_PRIORITY block. Then, it transmits its request message during the IPG and transmits the packet. However, if a higher priority node transmits request message in the same IPG interval, the lower priority node gives up transmitting and

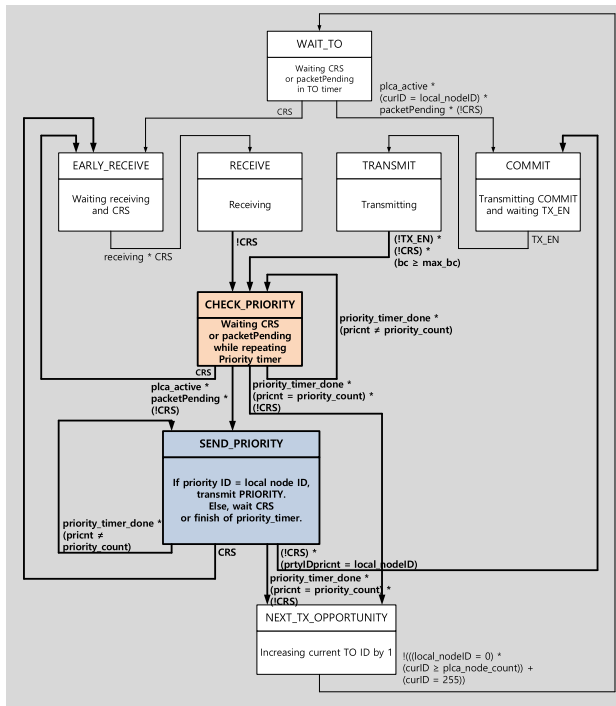


FIGURE 13. State diagram of the physical layer collision avoidance control.

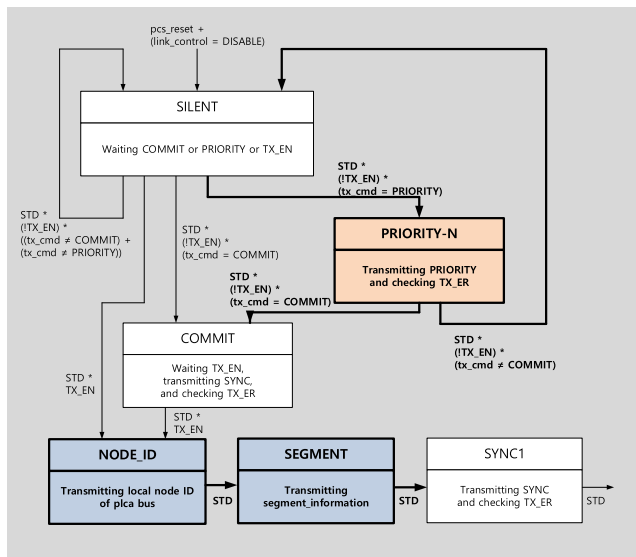


FIGURE 14. State diagram of the physical coding sublayer transmission.

receives packets from the higher one. If there is no packet to transmit, then the transmission opportunity is increased after waiting for all timers to end.

Fig. 14 illustrates the state diagram of the physical coding sublayer (PCS) transmission. Blocks in bold lines are included for processing the priority and segmentation. The priority packet request signal is transmitted before the COMMIT signal. Then, the node transmitting the COMMIT signal transmits its node ID in the NODE_ID block, and sends the

segment information in the SEGMENT block so that the receiving node checks if the received packet is segmented or is the last segment. Finally, the transmitting node starts sending the SYNC signal in the SYNC1 block.

The two algorithms described above, priority allocation and segmentation, may result in additional computational complexity. The transmitter needs to send the packets within the segment size, and the receiver should reassemble multiple packet segments. For this purpose, each node should own additional queues, count the packet length, and check the IDs of each received segments. Although this additional complexity should be considered in the design of the algorithm, it is expected that the advantage of latency reduction is large enough to compensate for the complexity in the bus topology of vehicular network.

IV. PERFORMANCE ANALYSIS

In this section, we analyze the performance of the proposed PLCA algorithm. The analysis parameters are described in Table 2, and MATLAB was used in the simulation. The number of nodes in the PLCA bus is eight, and the data rate is 10 Mbps. The beacon timer, which waits for a BEACON signal, is set to 20 bits, and the TO timer, which waits for the TO, is set to 32 bits. The lengths of COMMIT and PRIORITY are 5 bits each. The payload length of a priority node is set to 42 to 91 bytes considering the CAN-FD mapping in Fig. 4, where the CAN-FD frame has a maximum of 91 bytes, including bit stuffing. If the length of the priority packet is increased, then it will cause more delay at the lower priority nodes and regular nodes. However, the effect will not be considerable since the number of priority nodes is limited. The payload length generated at other regular nodes is defined as 42 to 1,500 bytes. Each node is configured to have a 100-Kbyte queue.

TABLE 2. Simulation parameters.

Parameter	value
Number of nodes	8
Bit rate	10 Mbps
Beacon timer	20 bits
COMMIT message	5 bits
PRIORITY message	5 bits
TO timer	32 bits
Payload of normal packet	42~1,500 bytes
Payload of priority packet	42~91 bytes [12]
Queue	100 Kbytes

Fig. 15 represents the pseudocode of the simulation, describing the data transmission process by PLCA, employing prioritization and segmentation. Each node generates 30,000 packets according to the Poisson distribution and updates the queue using those packets by the variable ‘current time’. Each node has a TO in a round-robin manner for T seconds. The node with the TO is called the ‘current node’, and the node with high priority is the ‘priority node’.

```

Begin
Initialize all nodes on plca bus
Generate packets at each node

While current Time is smaller than T sec then
  Update queue

  If current node has data then
    Transmit data
    Update current Time
    Update queue

    While priority-node has data then
      Transmit data
      Update current Time
      Update queue
    end while
  else
    Yield the Transmit_opportunity
  end if

  Transmit_opportunity++

  If Transmit_opportunity > the number of nodes then
    Transmit_opportunity = 1
  end if
end While
    
```

FIGURE 15. Pseudocode for the performance analysis.

If the current node transmits data, the delay is calculated, and the current time is updated considering this delay. Then, the queue is updated again. The priority node can transmit its packet by sending the priority request signal PRIORITY after the current node finishes its transmission. If there is no priority request, the TO is increased.

In this way, the performance of the proposed network is evaluated. The average delay is determined by dividing the sum of the delays of all successfully transmitted packets by the number of packets. The maximum delay is the largest one that these packets experience. The total load changed from 0.1 to 1.0 during the simulation, and the load was evenly distributed among all nodes in the bus.

Fig. 16 lists the simulation results when only prioritization is used with one priority node. The delay of the priority node is reduced, whereas the delay of the regular nodes is slightly increased compared to when priority is not applied. The average delay is 0.3 msec for regular nodes, whereas the maximum delay is about 4 msec when the load is 0.5. In the case of the priority node, the average delay is 0.2 msec, and the maximum delay is about 1 msec at a load of 0.5. The reference load of 0.5 is used because most vehicular networks are designed to operate below this value.

Fig. 17 presents the delay for the case with two priority nodes. The node with the first priority is indicated Priority-1, whereas that with the second priority Priority-2. For Priority-1, the average delay is 0.2 msec, and the maximum delay is a little over 1 msec at a load of 0.5. In the case of Priority-2, the average and maximum delays are slightly greater than those of Priority 1, but the performance is very close. These results are consistent with Eqs. (3), and (4), where the maximum latency for the priority-1 and priority-2

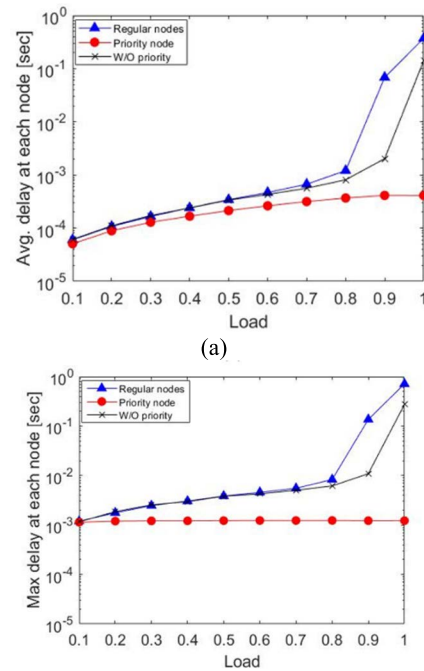


FIGURE 16. Delay of the PLCA when only prioritization with one priority node is used (a) average delay and (b) maximum delay.

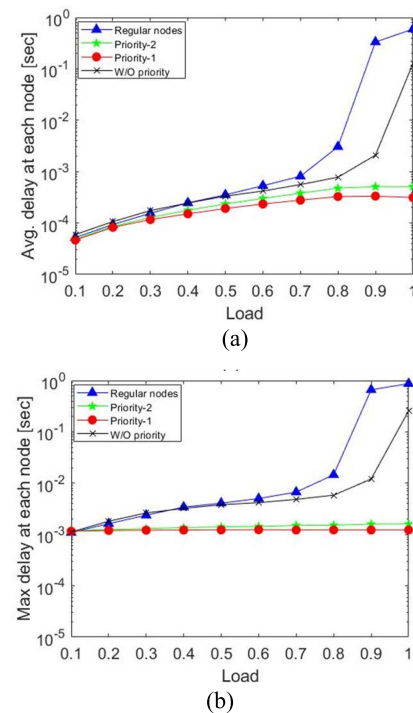


FIGURE 17. Delay of the PLCA when only prioritization with two priority nodes is used (a) average delay and (b) maximum delay.

node is 1.3 and 1.36 msec, respectively. For regular nodes in the simulation, the average delay is 0.36 msec and the maximum delay is about 4 msec when the load is 0.5. These results reveal that the proposed PLCA is effective for multiple delay-sensitive ECU nodes demanding delays of less than

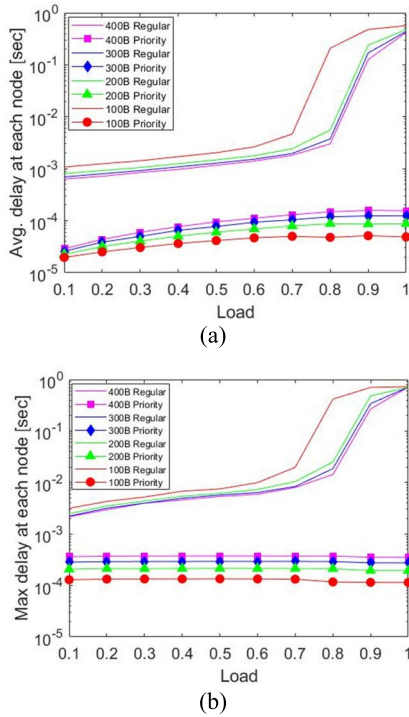


FIGURE 18. Delay of the PLCA with one priority node and packet segmentation (a) average delay and (b) maximum delay.

a few milliseconds. It can be used for HMI, comfort, and most safety electronics listed in Fig. 2 except for a few safety electronics and most engine/powertrain ECUs. Therefore, packet segmentation is introduced in this study to achieve further reduction in the packet delay.

Fig. 18 represents the network performance when packets are transmitted in different segment sizes in addition to the node prioritization. The delay is further reduced as the segment becomes smaller at the priority node, whereas it is the opposite for regular nodes. If the segment size is 100 bytes, data from the priority node have an average delay of 40 μ sec and a maximum delay of 130 μ sec at the load of 0.5, which can satisfy the requirements in most domains. In the case of regular nodes, the average delay is 2 msec, and the maximum delay is about 7 msec, which is a significant increase compared to that without segmentation. It occurs because long packets from the regular nodes are divided into several segments, which are transmitted in several cycles. In addition, the ratio of overhead parts compared to payload is increased.

Delays for the segment size of 200 bytes with two priority nodes are analyzed in Fig. 19. The priority nodes have an average delay of 75 μ sec and a maximum delay of 300 μ sec when the load is 0.5. The delays of the two priority nodes are quite close, and their maximum delays are reduced to less than one-third of the case without segmentation. Therefore, it is expected that this scheme can be used in most delay-sensitive ECUs.

Fig. 20(a) details the throughput of the proposed PLCA with one priority node for different segment sizes.

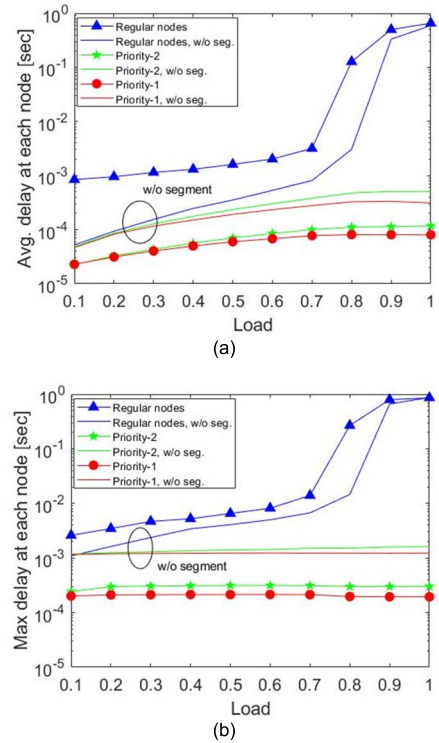


FIGURE 19. Delay of the PLCA with two priority nodes and packet segmentation for 200 bytes (a) average delay and (b) maximum delay.

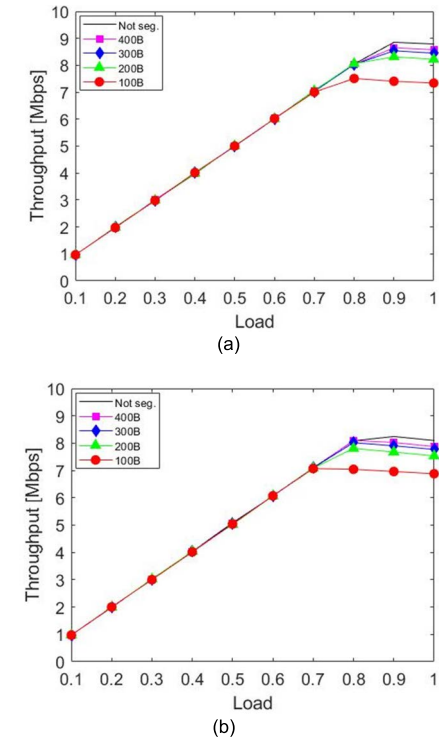


FIGURE 20. Throughput of the proposed PLCA for different packet segments (a) with one priority node and (b) with two priority nodes.

The throughput of the bus is saturated rapidly with the small segment size because a packet is divided into many segments, and each requires overhead parts. The throughput is

reduced further with two priority nodes in Fig. 20(b), which is attributed to each regular node having more packets with two priority nodes compared to the single priority case at the same load value since priority data have smaller packet size. However, the throughput is proportional to the load before a load of 0.7; thus, it is not degraded even after prioritization and segmentation. This outcome is attributed to the PLCA that prevents packet collision in the bus.

V. CONCLUSION

The 10BASE-T1S Ethernet using PLCA is a protocol that can support bus-type communication in a vehicle to replace the existing CAN communication. However, in its present form, it is challenging to use it in ECUs demanding a delay of less than a few milliseconds. Node prioritization and packet segmentation schemes are introduced in this study to solve this problem. The designated priority nodes can have a TO ahead of the regular nodes. If two priority nodes exist among the eight nodes in a bus, a maximum delay of 1 msec is estimated for the data from priority nodes. In addition, packet segmentation reduces the bus cycle by limiting the Ethernet packet size. Assuming a segment of 100 bytes and prioritization are used in the PLCA, the maximum delay for priority data is 130 μ sec. The segment size should be decided by considering the delay requirements for each node. This study found that the 10BASE-T1S can connect most delay-critical nodes in a vehicle if the PLCA function adopts the proposed prioritization and segmentation. This approach can replace or interface with the existing CAN or CAN-FD.

REFERENCES

- [1] P. Hank, S. Müller, O. Vermesan, and J. Van Den Keybus, "Automotive Ethernet: in-vehicle networking and smart mobility," in *Proc. Des., Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2013, pp. 1735–1739.
- [2] Ixia. *Automotive Ethernet: An Overview*. Accessed: Feb. 11, 2022. [Online]. Available: https://support.ixiacom.com/sites/default/files/resources/whitepaper/ixia-automotive-ethernet-primer-whitepaper_1.pdf
- [3] C. Correa, C. M. Kozierok, R. B. Boatright, and J. Quesnelle, "Automotive Ethernet: Definitive guide," *Intrepid Control Syst.*, pp. 175–176, 2014.
- [4] *IEEE Standard for Ethernet—Amendment 5: Physical Layer Specifications and Management Parameters for 10 Mb/s Operation and Associated Power Delivery over a Single Balanced Pair of Conductors*, IEEE Standard 802.3CG, 2019.
- [5] K. Matheus, "Evolution of Ethernet-based automotive networks: Faster and cheaper," *IEEE-SA Ethernet IP Automot. Technol. Day*, Oct. 2018.
- [6] D. Stohrmann, A. Kostrzewa, R. Ernst, and H. Kellermann, "Towards OFDMA-based Ethernet for future in-vehicle communication," in *Proc. 2nd Int. Conf. Societal Autom. (SA)*, May 2021, pp. 1–8.
- [7] G. W. D. Besten, "Single-pair automotive PHY solutions from 10 Mb/s to 10 Gb/s and beyond," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2019, pp. 474–476.
- [8] M. Sato, "Automobile network system," *Sungandang*, pp. 24–83, 2010.
- [9] R. Hegde, S. Kumar, and K. S. Gurumurthy, "The impact of network topologies on the performance of the in-vehicle network," *Int. J. Comput. Theory Eng.*, vol. 5, no. 3, pp. 405–409, Jun. 2013.
- [10] M. Park, M. Yoon, K. Na, and D. Kim, "A study on application of time-triggered Ethernet for vehicle network," *J. Inst. Webcasting Internet Telecommun.*, vol. 15, no. 6, pp. 79–88, Dec. 2015.
- [11] G. Lee, S. Do, and J. Yun, "Automotive Ethernet technology," *KICS Inf. Commun. Mag.*, vol. 62, vol. 33, pp. 54–62, 2015.

- [12] M. Cakir, T. Hackel, S. Reider, P. Meyer, F. Korf, and T. C. Schmidt, "A QoS aware approach to service-oriented communication in future automotive networks," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2019, pp. 1–8.
- [13] M. Ibrar, A. Akbar, S. R. U. Jan, M. A. Jan, L. Wang, H. Song, and N. Shah, "ARTNet: AI-based resource allocation and task offloading in a reconfigurable internet of vehicular networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 67–77, Jan./Feb. 2022.
- [14] M. Ashjaei, L. L. Bello, M. Daneshtalab, G. Patti, S. Saponara, and S. Mubeen, "Time-sensitive networking in automotive embedded systems: State of the art and research opportunities," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102137.
- [15] C. Gunther, *An Overview With Some Specific Applications: IEEE P802.3CG 10 Mb/s Single Twisted Pair Ethernet*, Standard TSN 802.3CG, 2017.
- [16] V. Iyer and M. Rentschler, *Future-Proofing PLCA for Priority-Based Transmit Ordering*, IEEE Standard 802.3CG-10SPE, 2018.
- [17] J. Min, S. Jeong, and Y. Park, "Performance evaluation of automotive PLCA Ethernet using CAN-over-Ethernet," *J. Korean Inst. Commun. Inf. Sci.*, vol. 1458, vol. 46, pp. 1449–1458, 2021.
- [18] W. Zeng, M. A. S. Khalid, and S. Chowdhury, "In-vehicle networks outlook: Achievements and challenges," *IEEE Commun. Surveys Tuts.*, vol. 1571, vol. 18, no. 3, pp. 1552–1571, 3rd Quart., 2016.
- [19] M. Rahmani, R. Steffen, K. Tappayuthpijarn, E. Steinbach, and G. Giordano, "Performance analysis of different network topologies for in-vehicle audio and video communication," in *Proc. 4th Int. Telecommun. Netw. Workshop QoS Multiservice IP Netw.*, Feb. 2008, pp. 179–184.
- [20] D. G. Hickman, "Speed up your calibration with CAN FD," in *Proc. Int. Conf. Commun.*, 2013, pp. 02.12–02.18.



JIHYEON MIN received the B.S. and M.S. degrees in electronic engineering from Kookmin University, South Korea, in 2020 and 2022, respectively, where she is currently pursuing the Ph.D. degree in electronic engineering. Her research interests include vehicular networks, digital communications, and FPGA implementation of digital electronics. She is the Student Member of Korea Information and Communications Society.



YOUNGIL PARK received the B.S. and M.S. degrees in electrical engineering from Seoul National University, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree from Texas A&M University, USA, in 1995. He worked at Korea Telecom, from August 1995 to February 1999, in the area of optical communication. He studied high-speed WDM transmission and fiber non-linearity in the fiber network at Korea Telecom. Based on his achievements in this research area, he received the Excellent Research Award from Korea Telecom, in 1997. In March 1999, he joined Kookmin University, Seoul, where he has been a Professor at the Electronics Engineering Department, and served as the Chairman for the EE Department, from 2009 to 2010. He has authored more than 140 journals and conference papers. His research interests include optical communication systems, visible light communication, and vehicular networks. He is the Winner of paper awards presented at the Photonics Conference Korea, in 2010 and 2011, and the KICS Conference, in 2011, 2013, and 2015. He is the Lifelong Member of Korea Information and Communications Society.

...