symmetry

MDPI

Article

FlexRay Static Segment Message Scheduling Based on Heterogeneous Scheduling Algorithm

Shuqing Li, Yujing Wu, Yihu Xu, Kaihang Zhang and Yinan Xu *

Department of Electronic & Communication Engineering, Yanbian University, Yanji 133002, China; 2023050046@ybu.edu.cn (S.L.); yjwu@ybu.edu.cn (Y.W.); xuyh@ybu.edu.cn (Y.X.); 2689332993@ybu.edu.cn (K.Z.)
* Correspondence: ynxu@ybu.edu.cn

Abstract: With the development of intelligent connected vehicles, higher demands are being placed on the capabilities of in-vehicle bus networks. Compared to traditional in-vehicle bus networks like Local Interconnect Network (LIN) and Controller Area Network (CAN), the FlexRay bus offers advantages such as high real-time performance and high transmission rates, making it the core technology of the new generation of in-vehicle bus networks. This study focuses on the phenomenon of bandwidth resource waste in the FlexRay bus and innovatively proposes the FlexRay Static Segment Heterogeneous Scheduling Algorithm (SHSA). The SHSA algorithm optimizes the message transmission performance of the FlexRay bus through heterogeneous allocation of communication channels and message scheduling methods. This study established a simulation experimental platform using the CANoe.FlexRay bus network simulation tool and conducted simulation experiments on the proposed algorithm. Experimental results show that the average bandwidth utilization of the SHSA algorithm is 72.5%, which is 20.91%, 51.14%, and 54% higher than that of the existing Heterogeneous Makespan-minimizing DAG Scheduler (HMDS), Message Packing Scheme, and Jitter-aware Message Scheduling-Simulated Annealing and Greedy Randomized Adaptive Search Procedure (JAMS-SG), respectively. This study provides technical support for message transmission in intelligent connected vehicles and enhances the communication efficiency of the in-vehicle FlexRay bus network.

Keywords: intelligent connected vehicles; FlexRay; static segment; message scheduling; bandwidth utilization

check for updates

Academic Editor: Sergei Odintsov

Received: 18 April 2025
Revised: 30 April 2025
Accepted: 1 May 2025
Published: 2 May 2025

Citation: Li, S.; Wu, Y.; Xu, Y.; Zhang, K.; Xu, Y. FlexRay Static Segment Message Scheduling Based on Heterogeneous Scheduling Algorithm. *Symmetry* **2025**, *17*, 696. https://doi.org/10.3390/sym17050696

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the increasing complexity and functional requirements of automotive electronic control systems, the demands for the timeliness and reliability of messages have been heightened, making the optimization of symmetry in in-vehicle communication systems particularly important. Symmetry holds significant importance in communication systems as it can enhance resource utilization, reduce latency, and strengthen system reliability [1,2]. LIN and CAN buses have limitations in data transmission speed and communication mechanisms. The LIN bus is a low-cost, low-complexity serial communication protocol, primarily used in body control networks for window control, seat adjustment, windshield wiper systems, and other applications [3]. The CAN bus is widely used for the transmission of critical data in vehicle sensor networks, power systems, and other applications [4,5]. The FlexRay bus, as a high-performance real-time communication protocol, features a high-speed transmission rate of 20 Mbps, high reliability, fault tolerance, and is used in automotive suspension and chassis control systems [6,7]. The FlexRay bus plays an important role in in-vehicle communication, as well as in intelligent connected vehicles such

as autonomous driving, advanced driver-assistance systems, and vehicle-to-everything communication [8,9].

The FlexRay bus is responsible for data transmission between Electronic Control Units (ECUs). Due to the increasing number of messages on the in-vehicle bus, the bus network is experiencing insufficient bandwidth, information delays, and data loss. These phenomena reflect the asymmetry that exists in the communication system between the supply of bandwidth and the demand for transmitting messages [10].

The research on FlexRay message scheduling mainly employs methods such as parameter optimization, frame packing, and message priority optimization to further improve network utilization. Kukkala et al. [11] propose a jitter-aware message scheduling framework. This study combines the optimization of message scheduling tables and jitter handling mechanisms to address time-triggered and high-priority event-triggered messages affected by jitter, minimizing the impact of jitter on the communication performance of the FlexRay bus. Ricardo et al. [12] propose a message scheduling method based on heuristic algorithms. By analyzing the time constraints of the static and dynamic segments of the FlexRay protocol and combining parameters such as message periods and deadlines, a software tool was developed to optimize message scheduling. Alsaidy et al. [13] propose a particle swarm optimization task scheduling algorithm based on heuristic initialization. By combining the Longest Job to Fastest Processor (LJFP) algorithm and the Minimum Completion Time (MCT) algorithm to initialize the particle swarm of the algorithm, it enhances the performance of task scheduling. Kumar et al. [14] propose a method to achieve bandwidth minimization in the static segment using response time analysis techniques. By analyzing the characteristics of the static segment of the FlexRay protocol and combining message frame packing and message scheduling issues, a fast algorithm is developed to optimize the message frame packing scheme, thereby minimizing the bandwidth utilization of the static segment.

Effective message scheduling algorithms are an effective means to reduce bandwidth resource waste. However, existing FlexRay message scheduling research, such as JAMS-SG [11] and the Message Packing Scheme [14], lacks message scheduling methods based on FlexRay communication channels, resulting in the underutilization of FlexRay's two communication channels. Meanwhile, algorithms like HMDS [15] focus solely on heterogeneous scheduling without providing efficient message scheduling. This study proposes a heterogeneous message scheduling algorithm for a static segment of FlexRay. Using a heterogeneous allocation method for static segment message communication channels and combining it with the Integer Linear Programming (ILP) message scheduling strategy, a message scheduling table for the FlexRay static segment is established. This algorithm achieves symmetry optimization in the communication system, ensuring a more balanced utilization of resources across the two communication channels. It avoids the over-concentration or waste of resources, thereby enhancing the overall system's resource utilization efficiency and communication performance. Section 2 introduces the FlexRay bus network. Section 3 proposes a heterogeneous message scheduling algorithm for the FlexRay static segment. Section 4 evaluates the performance of the SHSA algorithm. Finally, Section 5 summarizes the research content and draws conclusions.

2. FlexRay Bus Network

The FlexRay bus has two channels, each with a maximum data rate of 10 Mbps, allowing for a total data rate of up to 20 Mbps. The FlexRay node structure includes four main components: Host, Communication Controller, Bus Guardian, and Bus Driver, as shown in Figure 1. Among them, Node 1 and Node 2 are connected to Channel 1 and Channel 2, respectively, through their respective Bus Drivers and Bus Guardians [16].

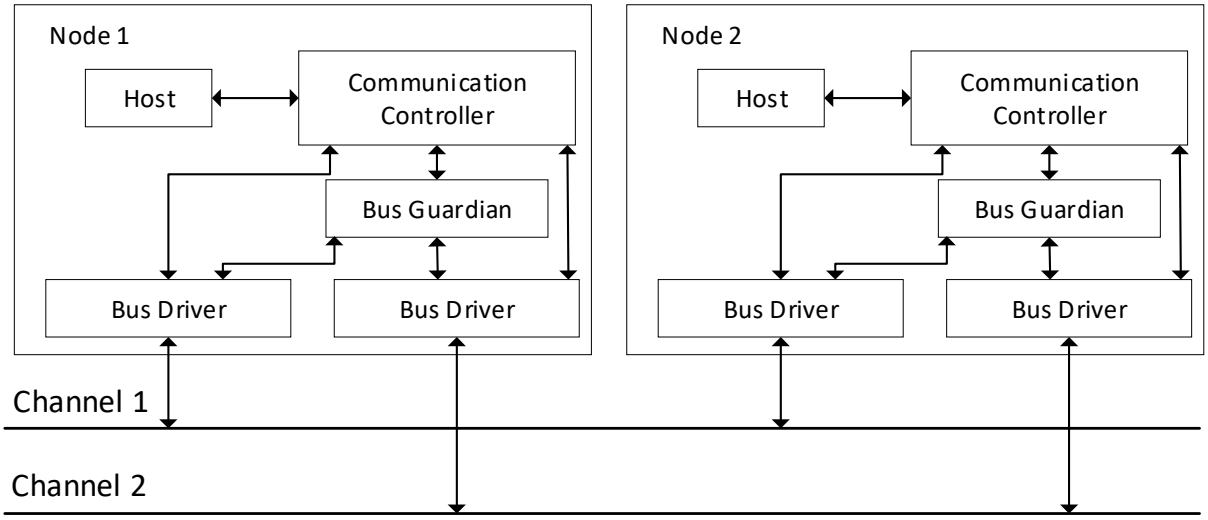


Figure 1. FlexRay node structure.

The FlexRay communication cycle consists of four parts: static segment, dynamic segment, symbol window, and network idle time [17,18], as shown in Figure 2. Among them, the slots in the static segment are fixed, while the minislots in the dynamic segment can be adjusted as needed. The symbol window is used for transmitting characteristic symbols, and the symbol window is crucial for the synchronization and configuration of the communication system. Network idle time is used for error detection and recovery in the communication system.

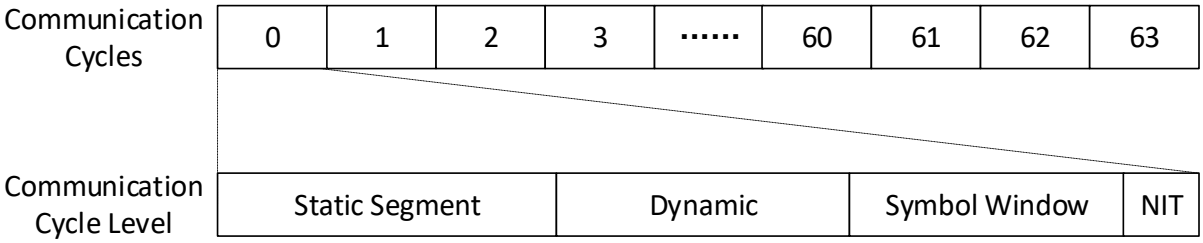


Figure 2. FlexRay communication cycle.

2.1. FlexRay Data Frame Structure

The FlexRay data frame consists of three parts: Header Segment, Payload Segment, and Trailer Segment [19], as shown in Figure 3. The first five bits of the Header Segment include one reserved bit and four indicator bits. Frame ID (FID) occupies 11 bits and is used to determine the transmission location of the frame. The Payload Length occupies seven bits to record the length of the Payload Segment. The Header Cyclic Redundancy Check (CRC) is 11 bits and is used to verify the correctness of the transmitted information. The Cycle count is six bits and is used to indicate how many cycles the current information has gone through in the network from the time it is triggered to the current time. In the static segment, the length of the Payload Segment is fixed and determined by configuration parameters, with a maximum of 254 bytes. In the dynamic segment, the length of the Payload Segment is variable, ranging from 0 to 254 bytes. In the static segment, the first 12 bytes of the Payload Segment are used for the network management vector. In the dynamic segment, the first two bytes are used as the message ID to define the content of the data segment. The Trailer Segment contains a single field, a 24-bit CRC for the frame.

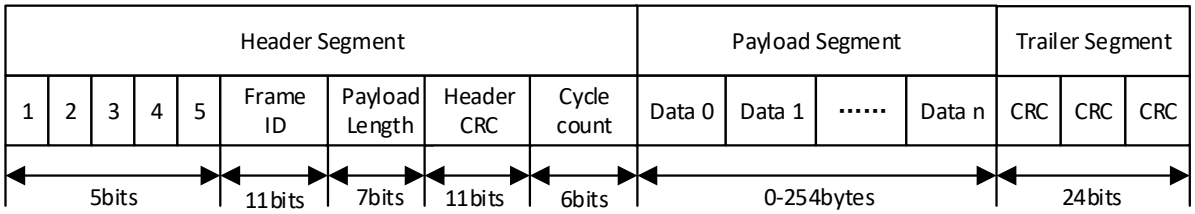


Figure 3. FlexRay frame format.

2.2. Transmission Characteristics of FlexRay Static Segment

The static segment of FlexRay uses the TDMA mechanism, providing highly deterministic real-time communication. The communication cycle of the static segment is divided into fixed-length slots, ensuring that each ECU sends data within the scheduled time, thereby avoiding conflicts and uncertainties during data transmission. The slot length of the static segment is determined based on the requirements of the maximum message length in the system, which helps ensure the timely transmission of critical data.

FlexRay data frames are divided into static frames and dynamic frames [20]. Among them, the static frame is an important unit for data exchange in the static segment. The FlexRay static frame encoding rules are shown in Figure 4. The static frame consists of three parts: Header, Payload, and Trailer. The header occupies five bytes. The payload occupies 0–254 bytes, used for transmitting valid information in the data frame. The trailer occupies three bytes. When the data in the static frame is transmitted in the static segment, a Transmission Start Sequence (TSS) needs to be added to initialize the starting point of the transmission sequence, occupying 3–15 bits. The Frame Start Sequence (FSS) occupies two bits. The Byte Start Sequence (BSS) is used to indicate the starting point of the byte. The Frame End Sequence (FES) is two bits long and follows the CRC in each frame. In addition, it is necessary to add the Communication Idle Delimiter (CID) and ActionPointOffset (APO). The APO is the starting time offset of each slot in the FlexRay static segment, used to determine the starting point of frame transmission. Its value can be configured according to system requirements, ranging from 1 to 63 macroticks. In the dynamic frame, it is necessary to add the Dynamic Trailing Sequence (DTS).

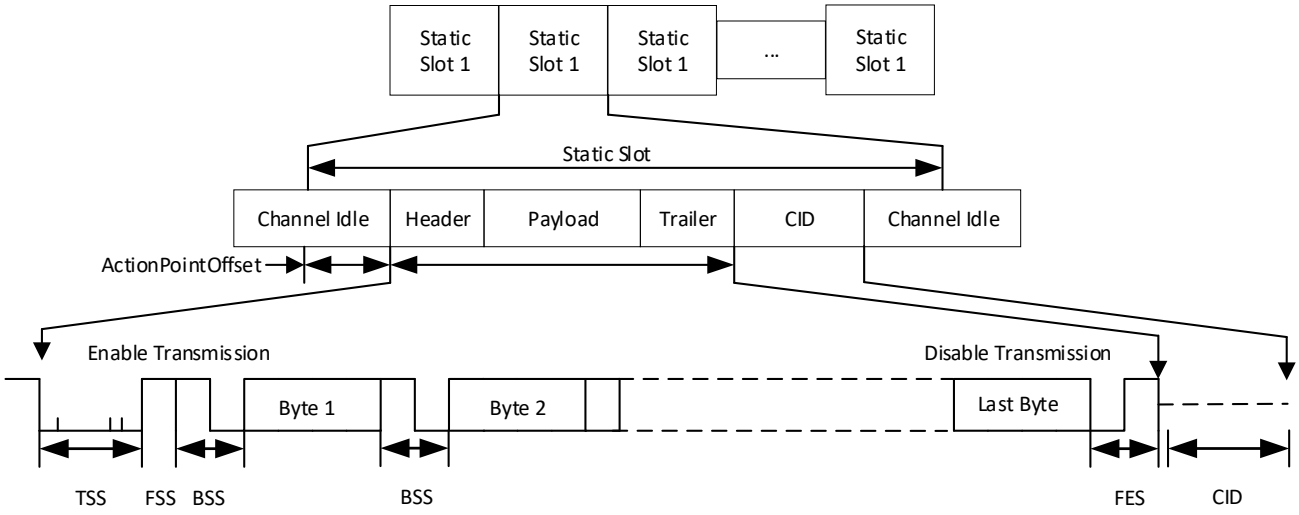


Figure 4. FlexRay static frame encoding rules.

Set the collection of messages for the static segment as $M = \{m_1, m_2, \ldots, m_n\}$. m_i is the i-th static segment message. When the length of m_i is w_i bits, the length of the message frame is $L(m_i)$, as shown by Formula (1).

$$L(m_i) = TSS + FSS + 80 + 20w_i + FES \tag{1}$$

The slot length of the FlexRay static segment is L_s, as shown by Formula (2):

$$L_s = 2 \times APO + ceil\left(((L(m_i) + CID) \times t_{bit} + MinPD + MaxPD)(t_{MT} \times (1 - CDM))^{-1}\right) \tag{2}$$

where *ceil* is the upper rounding function, t_{bit} is the transmission time of each bit, and $MinPD$ and $MaxPD$ are the minimum and maximum propagation delays, respectively. $MinPD$ is the minimum propagation delay, used to define the minimum time interval between two consecutive slots in the static segment. $MaxPD$ is the maximum propagation delay, used to define the maximum time interval between two consecutive slots in the static segment; t_{MT} is the time required to transmit a macrotick; and CDM is the maximum clock deviation, used to describe the maximum possible deviation range between node clocks.

For the j-th node e_j in the bus, the number of FIDs contained is FID_{e_j}. For a bus with n nodes, the total number of frame IDs required is N_{FID}, as shown by Formula (3).

$$N_{FID} = \sum_{j=1}^{n} FID_{e_j} \tag{3}$$

Bandwidth utilization reflects the proportion of payload in the total slot length within the static segment. The total slot length is fixed, while the payload length is related to the number and length of static frames. If there are too many static frames, the payload length of each frame will decrease, leading to reduced bandwidth utilization. Therefore, the purpose of FlexRay static segment scheduling is to ensure the maximization of bandwidth utilization while minimizing the number of FIDs.

3. Heterogeneous Message Scheduling Algorithm for Static Segment of FlexRay

This paper proposes an innovatively heterogeneous message scheduling algorithm for a static segment of FlexRay, which allocates messages to the two communication channels of the FlexRay bus, thereby achieving more symmetric resource utilization. Through the message scheduling strategy based on ILP, the algorithm further optimizes message scheduling to ensure that each message can be transmitted. This symmetry optimization not only improves bandwidth utilization but also reduces resource waste.

3.1. Architecture of SHSA Algorithm

The SHSA algorithm consists of heterogeneous allocation of communication channels, ILP message scheduling, and schedulability analysis, as shown in Figure 5.

First, obtain the set of messages M from the static segment of the FlexRay sender node, and extract parameters such as the length, period, and repetition for each message. Secondly, in the heterogeneous allocation of communication channels, messages are distributed to the two communication channels of the FlexRay bus. In ILP message scheduling, based on the integer linear programming strategy, the messages of each communication channel are scheduled to ensure that each message can be transmitted within the scheduled time. Finally, conduct a schedulability analysis and evaluation of the scheduling results. If the schedulability conditions are met, generate the message schedule table for the FlexRay static segment; if the schedulability conditions are not met, the process is concluded.

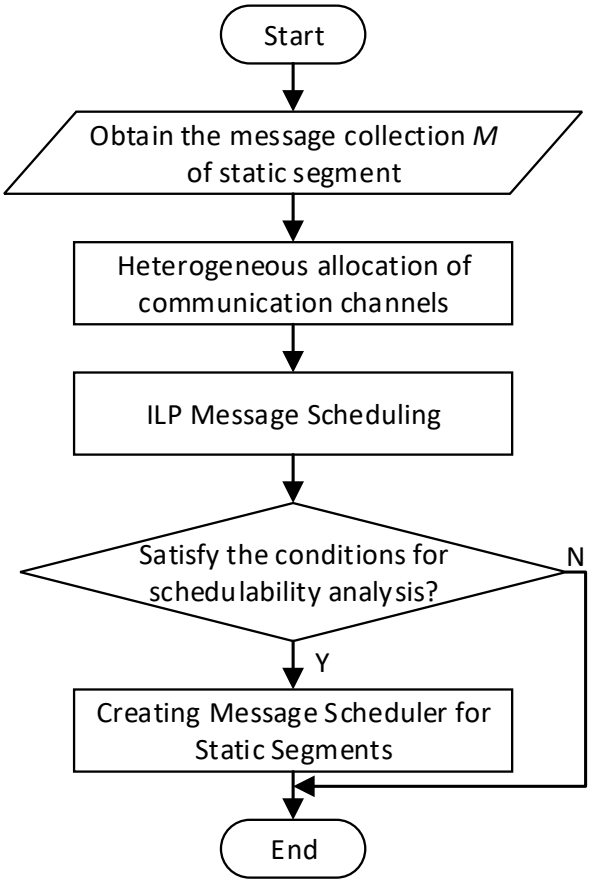


Figure 5. Flowchart of the SHSA algorithm.

3.2. Heterogeneous Allocation of Communication Channels

In order to fully utilize the two communication channels of the FlexRay bus and improve the bandwidth utilization, this paper proposes a heterogeneous allocation method for the communication channels. The method is divided into two parts: sorting and allocation of the message, as shown in Algorithm 1. First, the message set M of the static segment is obtained in the node of the FlexRay sender, and the function $Sort(M)$ is used to prioritize the messages in the message set M. The messages in the message set M are sorted in accordance with the principle of priority from highest to lowest, and the result is stored in the list (line 1). Calculate the communication time T_i for each message m_i in the message set M (line 2). Calculate the channel assignment parameter value $CH_A(m_i, CH_{1,2})$ for each message m_i on two channels. Message m_i is transmitted in Channel 1 if its channel assignment parameter in Channel 1 is greater than its channel assignment parameter in Channel 2, otherwise it is transmitted in Channel 2 (lines 3–10). Finally, the channel allocation result is returned (line 11).

During message transmission, the prioritization of messages has a significant impact on the scheduling results. A message with a high priority ensures the transmission of the message, while a message with a lower priority does not guarantee the transmission of the message. In Algorithm 1, the messages m_i in the message set M are sorted according to the order principle of decreasing repetition. Messages m_i with the same repetition are prioritized in decreasing order of payload length. The set of messages in the static segment is $M = \{m_1, m_2, \ldots, m_n\}$, and r_{m_i} is the repetition of message m_i, which is prioritized according to the principle of decreasing repetition. When r_{m_i} repetition is greater than or equal to r_{m_j}, the value of i is less than j, as shown by Formula (4).

$$\mathrm{r_{m_i}} \geq \mathrm{r_{m_j}} \; ? \; \mathrm{j} > \mathrm{i} : \mathrm{j} < \mathrm{i} \left(\forall m_{i,j} \in M, \; i \neq j \right) \tag{4}$$

Algorithm 1: Channel allocation algorithm

Input: Set of static segment messages M
Output: Channel allocation results
1: list Sort(M);
/*
Input: Set of static segment messages M
Output: Priority sorting result list. list sort M by descending order of priority;
Return list
/*
2: Calculate the communication time T_i for signal m_i in the static segment message set M;
3: **for** $m_i \in list$ **do**
4: calculate $CH_A(m_i, CH_{1,2}) = CH_A_{pre}(m_i, CH_{1,2}) + CH_A_{sub}(m_i, CH_{1,2})$;
5: **if** $CH_A(m_i, CH_1) \geq CH_A(m_i, CH_2)$ **then**
6: assign m_i to Channel 1;
7: **else**
8: assign m_i to Channel 2;
9: **end if**
10: **end for**
11: **Return** Channel allocation results

The set of messages ordered by repetition is defined as M', and when two messages have the same repetition, the message with a large payload is prioritized for transmission, as shown by Formula (5).

$$L_{m_i} \geq L_{m_j} \, ? \, \mathrm{j} > \mathrm{i} : \mathrm{j} < \mathrm{i} \Big(\forall m_{i,j} \in M', \; i \neq j, \; r_{m_i} = r_{m_j} \Big) \tag{5}$$

T_i is the communication time of message m_i, as shown by Formula (6), where L_{m_i} is the payload length of message m_i and R is the transmission rate of the FlexRay bus.

$$T_i = L_{m_i} / R \tag{6}$$

In order to utilize the communication channel of FlexRay efficiently, this paper gives a heterogeneous allocation method of the communication channel for message m_i. $CH_A(m_i, CH_{1,2})$ is the channel allocation parameter, as shown by Formula (7).

$$CH_A(m_i, CH_{1,2}) = CH_A_{pre}(m_i, CH_{1,2}) + CH_A_{sub}(m_i, CH_{1,2}) \tag{7}$$

$CH_{1,2}$ are two FlexRay communication channels CH_1 and CH_2. The channel allocation parameter $CH_A(m_i, CH_{1,2})$ is derived by iterating through the communication time T_i of message m_i. The minimum times required for message m_i to be allocated and transmitted on channels CH_1 and CH_2 are $CH_A(m_i, CH_1)$ and $CH_A(m_i, CH_2)$, respectively. By comparing the channel allocation parameters $CH_A(m_i, CH_1)$ and $CH_A(m_i, CH_2)$ for message m_i across the two channels, the channel with the smaller channel allocation parameter is chosen for the final transmission of message m_i.

$CH_A_{pre}(m_i, CH_{1,2})$ is calculated by iterating through the communication time T_i of the message $pre(m_i)$, which allows us to determine the time required for messages with a higher priority than m_i to complete transmission on channels CH_1 and CH_2, denoted as $CH_A_{pre}(m_i, CH_1)$ and $CH_A_{pre}(m_i, CH_2)$, respectively. This represents the time when message m_i begins to be transmitted on channels CH_1 and CH_2, as shown by Formula (8). $CH_A_{sub}(m_i, CH_{1,2})$ is determined by iterating through the communication time T_i

of the message $sub(m_i)$, which enables us to calculate the minimum time required for messages with a lower priority than m_i to be allocated and transmitted on channels CH_1 and CH_2, represented by $CH_A_{sub}(m_i, CH_1)$ and $CH_A_{sub}(m_i, CH_2)$, respectively, as shown by Formula (9), where $sub(m_i)$ denotes a message with a lower priority than m_i and $pre(m_i)$ is a message with a higher priority than m_i.

$$CH_A_{pre}(m_i, CH_{1,2}) = \underset{m_j \in pre(m_i)}{max} \left[CH_A_{pre}(m_j, CH_{1,2}) + T_j \right] \tag{8}$$

$$CH_A_{sub}(m_i, CH_{1,2}) = \underset{m_j \in sub(m_i)}{max} \left[\min \left\{ CH_A_{sub}(m_j, CH_{1,2}) + T_j \right\} \right] \tag{9}$$

3.3. ILP Message Scheduling

In this paper, we propose a message scheduling method based on ILP. The method achieves efficient message scheduling by utilizing linear programming. The message scheduling approach defines clear goals and constraints for each ECU, thus ensuring that messages are transmitted within the specified slots according to the periodicity requirements, while enhancing bandwidth utilization. Bandwidth utilization refers to the proportion of payload within the total slot length in the static segment. In the FlexRay bus, the total slot length is fixed, and the payload length is related to the number and length of static frames. By minimizing the number of static frames, the payload length of each frame will increase, thereby improving bandwidth utilization. Consequently, minimizing the number of FIDs can enhance the message transmission performance of the FlexRay bus. The ILP-related parameters are shown in Table 1.

Table 1. ILP-related parameters.

Parameter	Description
x_s	Binary variable, denotes whether slot s is allocated
$m_{s,b}$	Binary variable, denotes whether message m is allocated in time slot s with period b
S	Set of slots s
M	Set of messages m
E	Set of ECUs
r_m	Repetition of messages
L_s	Length of slots
L_m	Length of message m
k	periodic number
c	$c \in \{0, ..., k-1\}$ denotes period

The goal of ILP message scheduling is to minimize the number of FIDs, as shown by Formula (10).

$$min \sum_{e \in E} N_{FID} \tag{10}$$

During FlexRay message scheduling, the number of FIDs directly affects the number of occupied slots. The number of occupied slots is reduced by decreasing the number of FIDs, thereby increasing the network utilization of the FlexRay bus. With the objective of minimizing the number of FIDs, this paper proposes the following constraints so as to determine the slot and period of each message, as shown by Formulas (11)–(13).

$$\sum\nolimits_{s \in S} \sum\nolimits_{b=1}^{r} m_{s,b} = 1 \; \forall m \in M \tag{11}$$

$$\sum\nolimits_{m \in M} m_{s,mod(c \cdot r_m, 64)} \cdot L_m \leq L_s \; \forall m \in M, \; \forall s \in S, \; c \in \{0, \ldots, k-1\} \tag{12}$$

$$x_s \geq m_{s,b} \; \forall m \in M, \; \forall s \in S, \; b \in \{0, \ldots, r-1\} \tag{13}$$

Scheduling each message in a slot with a specific period is performed using Formula (11). Through Formula (12), it is ensured that the sum of the message lengths scheduled in a communication cycle must not be greater than the specified slot length. Through Formula (13), it is ensured that messages are scheduled in the allocated slots. The runtime of ILP message scheduling depends on the number of messages and the number of slots. The complexity is defined as $O(|M|\cdot|S|)$.

3.4. Schedulability Analysis

FlexRay static segments are time-triggered and mainly transmit messages with real-time and periodicity. When a group of messages needs to be transmitted, schedulability analysis is first performed to determine that all messages satisfy the scheduling conditions. Therefore, schedulability analysis is not only a core criterion for evaluating whether messages can be transmitted smoothly, but also an important indicator of message scheduling efficiency.

Message scheduling results are obtained by heterogeneous message scheduling algorithms, and subsequently message scheduling is analyzed based on constraints. The schedulability analysis has to satisfy the constraints of Formulas (14) and (15).

$$WCRT(m_i) \leq T_s \tag{14}$$

$$N_{FID} \leq N_s \tag{15}$$

where T_s is the duration of the slot, N_{FID} is the number of message FIDs, and N_s is the number of static segment slots. $WCRT(m_i)$, the maximum time required from the start of message transmission to the complete completion of message transmission, can be approximated by Formula (16).

$$WCRT(m_i) = (64 + L_{m_i})/R \tag{16}$$

where 64 bits is the sum of the frame header segment and the frame trailer segment., L_{m_i} is the payload length of message m_i and R is the transmission rate of the FlexRay bus. For each message m_i in the FlexRay static segment, to ensure that the message can be transmitted in the allocated slots, it must be satisfied that the $WCRT(m_i)$ of the messages allocated in each slot cannot exceed the duration of the slot, and that the number of FIDs does not exceed the number of configured static slots.

4. Simulation Experiments

CANoe is a powerful tool for bus network design, development, simulation, and testing developed by Vector in Germany. CANoe supports various bus protocols such as LIN, CAN, FlexRay, MOST, and Ethernet. CANoe.FlexRay is an extension module for the CANoe platform, dedicated to the development and simulation of the FlexRay communication protocol.

In this study, the topology of the in-vehicle network was built using the CANoe.FlexRay platform, as shown in Figure 6. Among them, the BLU node is used to send body control information, the Dashboard node is used to send dashboard information, the GearBox node is used to send transmission information, the BSC node is used to send vehicle braking information, and the Engine node is used to send engine operation status information. All of the above nodes are connected via the FlexRay bus FlexRay 1. CANoe.FlexRay is used as the main simulation platform, and hardware license support is provided through CANcaseXL, while VN8970 is used as the FlexRay communication interface, and the experimental environment is shown in Figure 7. To accurately reflect the

communication requirements and message characteristics of vehicles, this paper conducts simulations of message transmission between nodes based on the static segment messages of the FlexRay bus in real vehicles. The static segment message sets and their parameters are shown in Table 2.

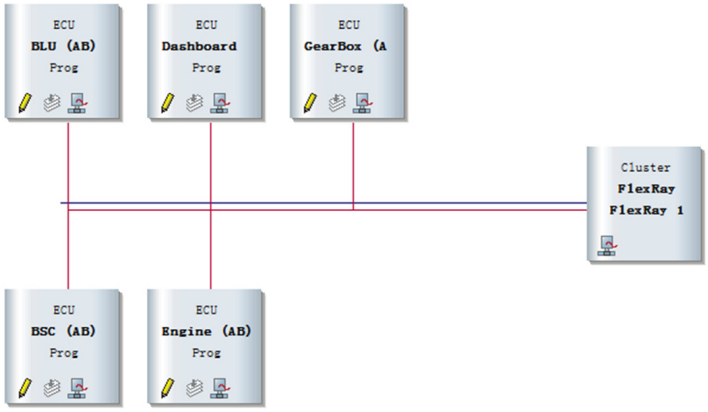


Figure 6. Topology of in-vehicle network.

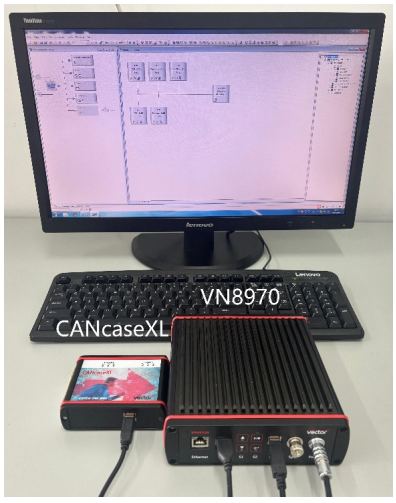


Figure 7. Experimental environment.

In order to validate the performance of the SHSA algorithm proposed in this study, a performance comparison with HMDS, Message Packing Scheme, and JASM-SG algorithms is carried out. Additionally, a performance comparison using the SHSA algorithm to schedule a message set affected by jitter has been added. The results of the bandwidth utilization comparison when the bus rate is 10 Mbps for each channel are shown in Figure 8. The experimental results show that the bandwidth utilization of the four algorithms is basically the same when the number of message frames is small. As the number of message frames increases, the SHSA algorithm proposed in this paper exhibits higher bandwidth utilization. In contrast, the HMDS, Message Packing Scheme, and JASM-SG algorithms have a lower bandwidth utilization growth rate as well as a lower bandwidth utilization than the SHSA algorithm ultimately achieved. The bandwidth utilization of the SHSA algorithm affected by jitter exhibits fluctuations, and the overall bandwidth utilization is also reduced. The results indicate that when transmitting the same number of static segment messages, the SHSA algorithm can more effectively utilize the bandwidth resources of the FlexRay bus, thereby enhancing transmission performance. However, jitter impacts the transmission performance of messages, leading to a decrease in bandwidth utilization.

Table 2. Static segment message sets and their parameters.

Message	Length (Byte)	Repetition	Message	Length (Byte)	Repetition	Message	Length (Byte)	Repetition
ABSWarninglamp	1	2	EngForce	16	2	New_Element 18441	16	2
AccelerationForce	16	2	EngPower	16	2	New_Element_19865	16	2
BackUpLight	1	2	EngSpeed	16	2	New_Element 59120	16	1
BatteryWarnLamp	1	2	EngTemp	7	2	New_Element 73321	16	1
BrakeLight	1	2	ErrorCode	6	2	New_Element_9074	16	2
BrakePressure	15	2	ESPWarningLamp	1	2	New_Element_9090	16	2
BrakeWarningLamp	1	2	SigDev4	16	1	oilWarningLamp	1	2
CarSpeed	16	2	Gear	3	2	PetrolLevel	8	2
CurGear	3	2	GearLock	1	2	ShiftRequest	1	2
Diagnostics	8	2	ASRMode	1	2	SigDev1	16	2
EcoMode	2	2	IdleRunning	1	2	SigDev2	16	1
EngForce	16	2	New_Element_17031	16	1	SigDev3	16	1
status	2	2	SigDev5	16	1	WaterWarnlamp	1	2

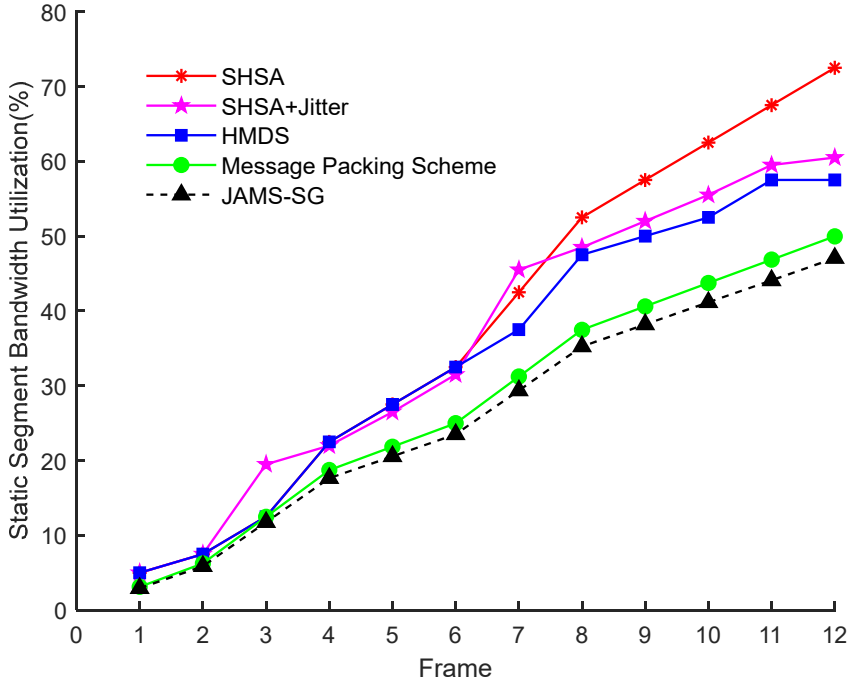


Figure 8. Bandwidth utilization comparison results.

Table 3 provides the bandwidth utilization of SHSA, HMDS, Message Packing Scheme, and JASM-SG algorithms. The experimental results show that the bandwidth utilization of the SHSA algorithm proposed in this paper is 72.5%. The bandwidth utilization of the SHSA algorithm is 20.91%, 51.14%, and 54% higher than that of the bandwidth utilization of the HMDS, Message Packing Scheme, and JASM-SG algorithms, respectively. Additionally, we have calculated the 95% confidence intervals for the algorithms. Specifically, the 95% confidence intervals for the SHSA, HMDS, Message Packing Scheme, and JAMS-SG algorithms are $35 \pm 8.1\%$, $30 \pm 7\%$, $27 \pm 6\%$, and $25 \pm 5.5\%$, respectively. These results clearly demonstrate that the SHSA algorithm proposed in this paper shows a significant performance improvement.

Table 3. Bandwidth utilization comparison.

Performance Indicators	SHSA	HMDS	Message Packing Scheme	JASM-SG
Bandwidth Utilization	72.5%	59.96%	47.97%	47.06%

5. Conclusions

This paper focuses on the message scheduling problem of the in-vehicle FlexRay bus and innovatively proposes a heterogeneous message scheduling algorithm for FlexRay static segments. The SHSA algorithm allocates communication channels for the transmitted messages through the heterogeneous allocation method of communication channels, and combines with the message scheduling method based on the linear programming of integers to schedule the messages. This improves the bandwidth utilization for message transmission and introduces a more symmetric and balanced communication model for the communication system. The experimental results show that the algorithm SHSA proposed in this paper enables the bandwidth utilization to reach 72.5%. Compared with traditional message scheduling algorithms, it not only significantly improves bandwidth utilization but also introduces a new message scheduling method by optimizing the symmetry of communication channels. This symmetry optimization paves new ways for the communication systems of intelligent connected vehicles and can be applied to the next-generation

in-vehicle communication systems to further optimize the bandwidth utilization of the FlexRay bus. However, the current SHSA algorithm does not address the scheduling of the FlexRay dynamic segment, and the SHSA algorithm is unable to handle dynamically changing messages. Future research can focus on extending this heterogeneous scheduling method from the static segment to the dynamic segment and deeply exploring its potential impact on overall network performance, thereby promoting the continuous advancement of in-vehicle communication technologies.

Author Contributions: Conceptualization, Y.X. (Yinan Xu) and S.L.; methodology, Y.X. (Yinan Xu); software, S.L.; validation, Y.X. (Yinan Xu) and S.L.; resources, Y.X. (Yinan Xu); data curation. S.L.; writing—original draft preparation, Y.X. (Yinan Xu) and S.L.; writing—review and editing, Y.X. (Yinan Xu); visualization, K.Z.; supervision, Y.W. and Y.X. (Yihu Xu); funding acquisition, Y.X. (Yinan Xu). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Jilin Province Natural Science Foundation, Grant Nos. YDZJ202501ZYTS641 and YDZJ202301ZYTS409, and was also supported by the National Natural Science Foundation of China, Grant Nos. 62161049 and 62201492.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are not publicly available due to privacy concerns. Requests for access to the data should be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Barletta, V.S.; Caivano, D.; Vincentiis, M.D.; Ragone, A.; Scalera, M.; Martín, M.Á.S. V-SOC4AS: A Vehicle-SOC for Improving Automotive Security. *Algorithms* **2023**, *16*, 112. [CrossRef]
2. Kim, H.-J.; Choi, M.-H.; Kim, M.-H.; Lee, S. Development of an Ethernet-Based Heuristic Time-Sensitive Networking Scheduling Algorithm for Real-Time In-Vehicle Data Transmission. *Electronics* **2021**, *10*, 157. [CrossRef]
3. Páez, F.; Kaschel, H. Design and Testing of a Computer Security Layer for the LIN Bus. *Sensors* **2022**, *22*, 6901. [CrossRef] [PubMed]
4. Chen, J.; Zuo, Q.; Xu, Y.; Wu, Y.; Jin, W.; Xu, Y. Study of Fixed Point Message Scheduling Algorithm for In-Vehicle Ethernet. *Electronics* **2024**, *13*, 2050. [CrossRef]
5. Hamed, A.; El-Kharashi, M.W.; Salem, A.; Safar, M. Two-Layer Bus-Independent Instruction Set Architecture for Securing Long Protocol Data Units in Automotive Open System Architecture-Based Automotive Electronic Control Units. *Electronics* **2022**, *11*, 952. [CrossRef]
6. Alexandre, S.R.; Daniel, H.P.; Renato, V.B.H.; Edison, P.F.; Carlos, E.P. Impact Analysis of Electrical Fast Transients on FlexRay In-Vehicle Networks. *IEEE Trans. Electromagn. Compat.* **2021**, *63*, 294–300.
7. Wu, Y.; Li, S.; Liu, S.; Xu, Y. Efficient Message Scheduling for FlexRay Dynamic Segments. *Symmetry* **2025**, *17*, 380. [CrossRef]
8. Lee, T.-Y.; Lin, I.-A.; Liao, R.-H. Design of a FlexRay/Ethernet Gateway and Security Mechanism for In-Vehicle Networks. *Sensors* **2020**, *20*, 641. [CrossRef] [PubMed]
9. Sugihara, M. Dynamic Slot Multiplexing Under Operating Modes for TDMA-Based Real-Time Networking Systems. *Electronics* **2020**, *9*, 224. [CrossRef]
10. Khatri, N.; Shrestha, R.; Nam, S.Y. Security Issues with In-Vehicle Networks, and Enhanced Countermeasures Based on Blockchain. *Electronics* **2021**, *10*, 893. [CrossRef]
11. Kukkala, V.K.; Bradley, T.; Pasricha, S. Reliable Real-Time Message Scheduling in Automotive Cyber-Physical Systems. *Mach. Learn. Optim. Tech. Automot. Cyber-Phys. Syst.* **2023**, *978*, 3–42.
12. Vaz, R.M.; Hodel, K.N.; Santos, M.M.D.; Arruda, B.A.; Netto, M.L.; Justo, J.F. An efficient formulation for optimization of FlexRay frame scheduling. *Veh. Commun.* **2020**, *24*, 100234. [CrossRef]
13. Alsaidy, S.A.; Abbood, A.D.; Sahib, M.A. Heuristic initialization of PSO task scheduling algorithm in cloud computing. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 2370–2382. [CrossRef]
14. Kumar, P.R.S.; Manjunatn, A.S.; Vinod, V. Efficient Utilization of Bandwidth in Static Segment of FlexRay Protocol. *SN Comput. Sci.* **2024**, *5*, 744. [CrossRef]

15. Senapati, D.; Sarkar, A.; Karfa, C. HMDS: A Makespan Minimizing DAG Scheduler for Heterogeneous Distributed Systems. *ACM Trans. Embed. Comput. Syst.* **2021**, *20*, 1–26. [CrossRef]
16. FlexRay Consortium. *FlexRay Communications System Protocol Specification, Version 3.0.1.*; FlexRay Consortium: Stuttgart, Germany, 2010; pp. 1–341.
17. Shen, Y.; Wang, Z.; Dong, H.; Lu, G.; Alsaadi, F.E. Distributed Recursive State Estimation for a Class of Multi-Rate Nonlinear Systems Over Wireless Sensor Networks Under FlexRay Protocols. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 1551–1563. [CrossRef]
18. Sunil, K.P.R.; Manjunath, A.S.; Vinod, V. Efficient handling of sporadic messages in FlexRay. *Perform. Eval.* **2024**, *166*, 102444.
19. Murvay, P.; Groza, B. Efficient Physical Layer Key Agreement for FlexRay Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9767–9780. [CrossRef]
20. Sunil, K.P.R.; Vinod, V.; Manjunath, A.S. Average delay analysis of soft deadline messages scheduled in the dynamic segment of FlexRay protocol. *Perform. Eval.* **2024**, *164*, 102404.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.