

FlexRay Static Segment Message Scheduling Optimization (Postprint)

Authors: Chen Shan, Shang Lihui, Zhang Fengdeng

Date: 2018-11-29T00:00:00+00:00

Abstract

To improve the bandwidth utilization of the static segment in FlexRay networks, this study investigates the optimization of the FlexRay static segment based on time-slot multiplexing. Signals to be transmitted are encoded into message frames, and the encoding problem is transformed into a constrained bin packing problem in the mathematical sense. Signals with transmission periods exhibiting a multiple relationship are encapsulated into a single message frame, and the BFD algorithm is employed to solve it. Based on the encapsulated message frames, a scheduling method is subsequently proposed to maximize static segment bandwidth utilization and minimize the number of frame IDs. Finally, the proposed method is validated on a FlexRay chassis integrated control system. The results demonstrate that the algorithm enhances bandwidth utilization by 18.7%, reduces the number of FIDs occupied per communication cycle by 90.47%, and improves static slot utilization within the cycle period by 41.52%.

Full Text

Preamble

Vol. 37 No. 1

Application Research of Computers (ChinaXiv Cooperative Journal)

Research on Scheduling Optimization of FlexRay Static Segment Messages

CHEN Shan, SHANG Lihui, ZHANG Fengdeng

School of Optical-Electrical & Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

Abstract: To improve the bandwidth utilization of the FlexRay static segment, this paper investigates optimization based on timeslot multiplexing. The

transmitted signals are encoded into message frames, transforming the encoding problem into a constrained bin packing problem in the mathematical sense. Signals with periods in integer multiple relationships are encapsulated into a single message frame and solved using the BFD algorithm. Based on the encapsulated message frames, a scheduling method is proposed to maximize static segment bandwidth utilization and minimize the number of frame IDs. The method is validated on a FlexRay chassis integrated control system. Results demonstrate that the algorithm increases bandwidth utilization by 18.7%, reduces the number of FIDs occupied per communication cycle by 90.47%, and improves static timeslot utilization within the cycle by 41.52%.

Keywords: FlexRay; bandwidth utilization; scheduling; timeslot multiplexing

0 Introduction

With increasing data communication volumes, traditional in-vehicle bus systems can no longer meet automotive network transmission requirements for reliability and determinism. To address these needs, the FlexRay multiplex real-time transmission network was developed, featuring high throughput, determinism, fault tolerance, and flexibility [1], and is expected to become the mainstream of future in-vehicle network development following CAN [2]. In real-time systems, scheduling aims to ensure that every task meets its timing constraints and responds to external requests in a timely and accurate manner [3]. Scheduling algorithms play a crucial role in network performance.

Current research on FlexRay scheduling algorithms primarily focuses on bandwidth optimization, with less attention paid to network security and reliability, while also overlooking critical aspects such as system scalability and communication real-time performance [4]. Literature [5] analyzed the transmission rules of static segment messages and provided methods for calculating message frame length and static timeslots. Literature [6] divided the message scheduling problem into signal packing and message timeslot allocation, using integer linear programming for modeling and solving with the optimization objectives of maximizing network bandwidth utilization and minimizing the number of timeslots. Literature [7] proposed a message scheduling strategy to minimize the required frame IDs while meeting schedulability constraints.

This paper focuses on static segment scheduling, conducting optimization research on static segment message scheduling schemes based on timeslot multiplexing mechanisms while satisfying real-time requirement time parameter configurations and improving static segment bandwidth utilization, with consideration for system scalability. Validation is performed on a FlexRay chassis integrated control and safety system, optimizing relevant parameters in the configuration system, analyzing scheduling optimization results, and conducting real-time performance evaluation.

1 FlexRay Communication Protocol

FlexRay provides advanced communication technology for in-vehicle applications, with a maximum data rate of 10 Mbps and total data rate up to 20 Mbps. When using redundant channel communication, it provides transmission fault tolerance; when using non-redundant channel communication, it improves bandwidth. The entire protocol is based on a time-triggered architecture, enabling deterministic communication [8–10]. A FlexRay node consists of five main components: Host (HOST), power supply, bus guardian (BG), bus driver (BD), and communication controller (CC).

1.1 Frame Format and Encoding

The static segment message frame consists of three parts: header segment, payload segment, and trailer segment. Its frame format is shown in [Figure 1: see original paper]. During transmission, nodes transmit data from left to right as illustrated in the figure. The static segment comprises 1 to $cSlotIDMax$ static timeslots, each composed of several macroticks, with all nodes in the network having identical static timeslot lengths.

The first five bits of the header segment are reserved bit, payload preamble indicator, null frame indicator, sync frame indicator, and startup frame indicator, respectively. The 11-bit frame ID defines the transmission timeslot for the message frame, specifying which timeslot in each communication cycle the frame is transmitted. The 7-bit payload length field indicates the payload size. The 11-bit header CRC validates header data integrity. The 6-bit cycle count field tracks the communication cycle number. The payload length is configurable but must be between 0 and 254 bytes. The trailer contains a 3-byte CRC checksum.

When performing scheduling optimization focusing on the payload segment, if a message frame's payload needs to transmit i bytes, the minimum required timeslot duration is:

$$T_{STS}(i) = \left\lceil \frac{109 + 10 \times i \times T_{gdBit}}{T_{MT}} \right\rceil \cdot T_{MT}$$

where the transmission start sequence is 15 bits; T_{gdBit} represents the time occupied by one bit, i.e., $T_{gdBit} = 1/C$, where C is the FlexRay bandwidth, typically taking values of 1, 2.5, 5, or 10 Mbps.

The static timeslot length is set to the longest message frame length in the network:

$$T_{STS} = \max_{m \in M} \{T_{STS}(i_m)\}$$

where i_m denotes the payload length of message m .

Due to length variations among message frame payloads, to avoid significant bandwidth waste and subsequent message transmission delays, static timeslot length must be optimally configured. If there are n periodic message frames (m_1, m_2, \dots, m_n) with corresponding payload lengths (i_1, i_2, \dots, i_n) bytes, the minimum length for transmitting these message frames is $\sum_{k=1}^n T_{STS}(i_k)$. From equation (2), the static segment bandwidth loss L is:

$$L = T_{STS} \times \sum_{k=1}^n f_k - \sum_{k=1}^n T_{STS}(i_k)$$

where f_k is the transmission frequency of each message frame within the cycle.

Network utilization [5] can be expressed as:

$$U = \frac{\sum_{k=1}^n T_{STS}(i_k)}{T_{STS} \times \sum_{k=1}^n f_k}$$

Analysis of equation (4) reveals that reasonable static timeslot configuration is critical for effective message scheduling. Building upon the above message frames, the concept of bandwidth utilization is introduced, characterizing how much allocated bandwidth is used for transmitting valid periodic data. Here, valid data refers to signal data constituting the message frame payload, leading to the concept of signal encapsulation into frames [11–13], where signal data is the actual data processed by each node's host.

Assume a network with N nodes, where $N = \{1, \dots, N\}$. For each node $n \in N$, there exists a signal data set $S_n = \{s_{n1}, s_{n2}, \dots, s_{nb_n}\}$. Each signal $s_{nb} \in S_n$ has three attributes: signal period p_{nb} , signal deadline d_{nb} , and signal data length b_{nb} (in bits). Since all signals must be scheduled within integer multiples of the communication cycle T_c , and ignoring dynamic segment signals, symbol window, and network idle time, the signal period T_c is selected as the greatest common divisor of all signal periods or its divisor, denoted as $T_{c,STS}$. In theoretical analysis, assuming $d_{nb} = p_{nb}$, this means only signals with identical periods can be encapsulated into one message, while also imposing constraints on message encapsulation.

For any signal s_{nb} in node n , the required bandwidth occupancy rate is:

$$D_{nb} = \frac{b_{nb}}{p_{nb}} \cdot \frac{1}{C}$$

The bandwidth occupancy rate for all signal data across the network is:

$$D = \sum_{n=1}^N \sum_{b=1}^{b_n} D_{nb}$$

For a message m_{nl} encapsulating signals from node n , the required bandwidth occupancy rate is:

$$A_{nl} = \frac{\sum_{s_{nb} \in \text{pack}(M_{nl})} b_{nb}}{p_{nl}} \cdot \frac{1}{C}$$

The bandwidth occupancy rate for all messages across the network is:

$$A = \sum_{n=1}^N \sum_{l=1}^{l_n} A_{nl}$$

Static segment bandwidth utilization U can be expressed as:

$$U = \frac{D}{A} = \frac{\sum_{n=1}^N \sum_{b=1}^{b_n} \frac{b_{nb}}{p_{nb}}}{\sum_{n=1}^N \sum_{l=1}^{l_n} \frac{\sum_{s_{nb} \in \text{pack}(M_{nl})} b_{nb}}{p_{nl}}}$$

1.2 Static Segment Communication Mechanism

FlexRay's communication cycle includes static segment (SS), dynamic segment (DS), symbol window (SW), and network idle time (NIT). The static segment employs a time division multiple access (TDMA) mechanism based on communication cycle cycles for message transmission. The static segment is divided into several equal-sized static timeslots with unique frame IDs. The frame identifier is transmitted along with other data, and by comparing it with the timeslot counter, it determines whether the message frame should be transmitted in the current timeslot.

1.3 Static Segment Transmission Characteristics

Since asynchronous systems enable software reuse, this research focuses on message and task scheduling in asynchronous systems, with message scheduling optimization based on timeslot multiplexing mechanisms. From the frame format, each payload segment data requires corresponding header and trailer additions, and after encoding, data is transmitted in real-time on the bus.

Bytes 0–12 of the static frame payload can serve as network management vectors, with other data transmitted in the remaining bytes, as shown in [Figure 2: see original paper].

Static frame encoding rules are illustrated in [Figure 3: see original paper]. Encoding adds transmission start sequence (TSS), frame start sequence (FSS), byte start sequence (BSS), and frame end sequence (FES).

The optimization objective is to minimize static timeslot length T_{STS} . For specific network data, blindly reducing static timeslot length adds extra bits during

encoding, preventing full bandwidth utilization for valid signal data transmission and increasing the number of message frames, thus requiring more static timeslot FIDs. The protocol stipulates that the number of static timeslots is limited, with a maximum of 1,023. For future expansion needs, the number of required FIDs must be minimized. The scheduling optimization objective is to find a proper balance under these constraints.

Before scheduling, the network is constrained as follows: a) No message transmission errors or failures exist in the FlexRay communication system; b) The period, deadline, and signal data length of all signals requiring transmission are known, assuming signal deadline equals its period to facilitate signal encapsulation; c) Communication cycle T_c is set as the greatest common divisor of all signal periods.

Within the same node, signals with different periods are allowed to be encapsulated into one frame, which may increase static timeslot length but reduces the number of used timeslots, benefiting future network expansion. This paper's optimization research on the static segment primarily converts to signal encapsulation optimization and FID allocation optimization.

2 Static Segment Message Scheduling Optimization

2.1 Signal Encapsulation Optimization

The purpose of signal encapsulation optimization is to maximize bandwidth utilization, i.e., solve for the minimum message bandwidth occupancy rate. Assume a FlexRay network with N nodes, where for any node n there exists $n \in N$, and the period set of signals contained in the node is $P_n = \{p_{n1}, p_{n2}, \dots, p_{nr_n}\}$. The set of signals with period p_{nj} is $S_{nj} = \{s_{nj1}, s_{nj2}, \dots, s_{njb_{nj}}\}$. For any signal $s_{njb} \in S_{nj}$, it contains three characteristics: signal period p_{njb} , data length b_{njb} , and deadline d_{njb} . Ideally assuming signal period equals its deadline, after specific encoding and optimization processing, the signal data yields a message frame set $M_n = \{m_{n1}, m_{n2}, \dots, m_{nl_n}\}$ with period set $P_{M_n} \subseteq P_n$. The set of messages with period p_{nk} is $M_{nk} = \{m_{nk1}, m_{nk2}, \dots, m_{nkl_{nk}}\}$.

Drawing on the x-group concept from typical FID allocation scheduling algorithms, this paper proposes allowing signals with integer multiple relationships within the same node to be encapsulated into one message frame, with the encapsulated message period being the least common multiple of all signal periods. The mathematical formulation of signal encapsulation processing is as follows:

Define two binary variables $X_{njp_i, p_{ml}}$ and $Y_{njp_i, p_{ml}}$. When the greatest common multiple of signal period p_{njb} and message period p_{nk} is p_{njk} , and signal s_{njb} is encapsulated into m_{nkl} , then $X_{njp_i, p_{ml}} = 1$; otherwise it is 0, as expressed in equation (11). When message m_{nkl} with period p_{nk} exists in a signal, then $Y_{njp_i, p_{ml}} = 1$; otherwise it is 0, as expressed in equation (12).

The mathematical description of the encapsulation method is:

$$\sum_{l=1}^{l_{nk}} X_{njp_i, p_{ml}} = 1, \quad \forall n \in N, \forall p_{nj} \in P_n, \forall s_{njb} \in S_{nj}$$

$$\sum_{b=1}^{b_{nj}} X_{njp_i, p_{ml}} \cdot b_{njb} \leq l_{STS}, \quad \forall n \in N, \forall p_{nk} \in P_{M_n}, \forall m_{nkl} \in M_{nk}$$

$$\sum_{b=1}^{b_{nj}} X_{njp_i, p_{ml}} \cdot b_{njb} \leq l_{STS}, \quad \forall n \in N, \forall p_{nk} \in P_{M_n}$$

Equation (13) indicates that the i -th signal with period p_{nj} can only be encapsulated into one message. Equation (14) represents that the total length (payload length value) of all signals encapsulated into message m_{nkl} must be less than the static timeslot length. Equation (15) imposes constraints on feasible static timeslot lengths based on protocol specifications and encapsulation methods, helping reduce iteration times.

Under these constraints, the overall network optimization objective is:

$$\min_{T_{STS}} \left(\sum_{n=1}^N \sum_{l=1}^{l_n} \frac{\sum_{s_{njb} \in \text{pack}(M_{nl})} b_{njb}}{p_{nl}} \right)$$

Since the bin packing problem is a discrete combinatorial optimization problem and a typical NP-complete problem [15–18], various approximation algorithms have been proposed. This design primarily employs the Best Fit Decreasing (BFD) algorithm to complete periodic signal encapsulation processing for FlexRay networks.

The BFD algorithm is suitable for offline scheduling of the static segment, completely determining each signal's transmission sequence during the design phase, with a worst-case asymptotic performance ratio of 11/9. Combining the above bin packing problem description, signal data length is treated as object volume, and static timeslot length as bin capacity. For each node in the FlexRay network, under discrete static timeslot lengths, signals with identical periods are prioritized for encapsulation into one message frame, and signals with integer multiple relationships are allowed to be encapsulated into the same frame.

Based on the mathematical model, the corresponding signal encapsulation optimization algorithm steps are:

- a) For each node n , first classify signals by period in ascending order, and within each period, arrange signals by length value in descending order.

- b) Determine static timeslot lower and upper bounds. The lower bound T_{STS}^L is shown in equation (17), representing the longest transmission time required among all signals without encapsulation. The upper bound T_{STS}^U is shown in equation (18), representing the longest transmission time required among all message frames when all integer multiple relationship messages are encapsulated into one frame.
- c) Through iteration for each static timeslot T_{STS} , increase by 2 bytes each time, perform the above encapsulation processing, and calculate bandwidth utilization values.
- d) Select the maximum value from all bandwidth utilization values obtained and extract the corresponding optimal static timeslot length and encapsulation method.

2.2 FID Allocation Optimization

Based on completed signal encapsulation, where the set of all messages and static timeslot length are known, timeslot multiplexing is introduced—where the same timeslot number can be allocated to different nodes in different cycles—to more effectively utilize static timeslots. Most studies have not provided practical static schedule tables. This design addresses these shortcomings by designing reasonable and effective static schedule tables to optimize static segment message scheduling.

For all encapsulated messages, they are represented as $m(n_i, p_j, c)$, where n_i ($i = 1, \dots, N$) denotes the node number, P_j denotes the message period, and c ($c = 1, \dots, r(p_j)$) denotes the message number under a specific node and period. A quadruple variable (n_i, f_m, r_m, b_m) is introduced to uniquely determine which node transmits which period's signal in a specific timeslot.

FID optimization allocation under timeslot multiplexing optimizes the total number of FIDs used in the network, allocating timeslots rationally from the message period perspective to utilize each timeslot more efficiently. Assume a FlexRay network has two nodes n_1, n_2 with message sets $m(n_1, 1, 1), m(n_1, 2, 1), m(n_1, 3, 1), m(n_2, 1, 1), m(n_2, 2, 1), m(n_2, 6, 1)$. Classical optimization scheduling would require 5 static timeslots. Applying timeslot multiplexing by arranging schedules in ascending period order reduces the required timeslot number to 4, as shown in .

TABLE:1 Static schedule instance

FID=1	FID=2	FID=3	FID=4
$m(n_1, 1, 1)$	$m(n_2, 1, 1)$	$m(n_1, 2, 1)$	$m(n_1, 3, 1)$
$m(n_1, 1, 1)$	$m(n_2, 1, 1)$	$m(n_2, 2, 1)$	$m(n_2, 6, 1)$
$m(n_1, 1, 1)$	$m(n_2, 1, 1)$	$m(n_1, 2, 1)$	
$m(n_1, 1, 1)$	$m(n_2, 1, 1)$	$m(n_2, 2, 1)$	$m(n_1, 3, 1)$

The one-dimensional bin packing problem description: Given a bin capacity c and a sequence of items $L = \{a_1, a_2, \dots, a_n\}$ where item size $s(a_i) \leq c$, the bin packing problem seeks a loading method A that partitions L into disjoint subsets B_j satisfying $\sum_{a_i \in B_j} s(a_i) \leq c$, with the total number of bins m being the minimum integer meeting this requirement [14].

The linear programming mathematical description of the bin packing problem is:

$$\begin{aligned} \min \quad & \sum_{j=1}^n y_j \\ \text{s.t.} \quad & \sum_{i=1}^n s_i x_{ij} \leq c y_j, \quad j = 1, 2, \dots, n \\ & \sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \\ & y_j = \begin{cases} 1, & \text{if the } j\text{-th bin is used} \\ 0, & \text{otherwise} \end{cases} \\ & x_{ij} = \begin{cases} 1, & \text{if item } i \text{ is placed in bin } j \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

Assume a FlexRay network contains x_c groups. All messages under each class form corresponding set L_x . For each message $x_m \in L_x$, its cycle period satisfies $j_m p_r T_c \leq 64$. The minimum number of FIDs required for this network is:

$$F_A = \sum_{x=1}^{x_c} \left\lceil \frac{\sum_{x_m \in L_x} j_m}{64} \right\rceil$$

The specific steps of the FID optimization algorithm without precedence constraints are:

- Calculate T_c (greatest common divisor of all message periods) and NTC (least common multiple of all message periods divided by T_c).
- Identify prime number periods in the message set, classify each message as described above, and sort elements in each x-group message set L_x by period and node number in ascending order.
- Apply corresponding algorithms to reasonably allocate f_m and b_m for each message, thereby assigning specific position variables (n_i, f_m, r_m, b_m) for each message. Assume u represents the timeslot utilization under NTC for a specific timeslot number, with cycle number $r_m \in \{0, 1, \dots, NTC-1\}$ and current timeslot number f_c . Initialize u and f_c to 0.

Assume a FlexRay network with two nodes n_1, n_2 containing identical messages, with the message set shown in .

TABLE:2 Message collection (i=1,2)

$m(n_i, 1, 1) | m(n_i, 1, 2) | m(n_i, 2, 1) | m(n_i, 2, 2) | m(n_i, 2, 3) | m(n_i, 3, 1) |$
 $m(n_i, 3, 2) | m(n_i, 3, 3) | m(n_i, 3, 4) | m(n_i, 4, 1) | m(n_i, 4, 2) | m(n_i, 4, 3) |$
 $m(n_i, 4, 4) | m(n_i, 4, 5) | m(n_i, 4, 6) | m(n_i, 4, 7) | m(n_i, 6, 1) | m(n_i, 4, 2) |$

Analysis of shows $T_c = 1$ ms and $NTC = 12$. Classical scheduling methods would require 16 static timeslots, with many wasted timeslots in different communication cycles. The proposed optimized scheduling scheme uses only 15 timeslots, demonstrating feasibility in improving static segment timeslot utilization and scheduling all messages with fewer timeslots, beneficial for future network expansion such as adding nodes or increasing message quantities.

3 Simulation Analysis

Based on FlexRay transmission rules analysis, this paper proposes an optimization method for FlexRay network static segment bandwidth utilization. Since the proposed method performs better on networks with larger data volumes and more nodes, validation is conducted on an actual automotive chassis integrated control system. The chassis integrated control system consists of Electronic Stability Control (ESC), Adaptive Cruise Control (ACC), FlexRay-CAN Gateway (GW), Electric Power Steering (EPS), Continuous Damping Control (CDC), and Air Suspension Control (ASC). The system's bus architecture is shown in [Figure 4: see original paper].

Each subsystem corresponds to a different node. The system's static message set [19] is shown in .

TABLE:3 Periodic static message collection

Node	Period (ms)	Length (byte)	Signal Representation
ESC	1	4	ESC-Status
ESC	2	6	ESC-WheelRot
ESC	4	16	ESC-WheelSpeed
ESC	6	1	ESC-DampReq
ACC	1	4	ACC-Status
ACC	2	6	ACC-ESCReq
GW	1	8	GW-Status
GW	16	1	EMS-AirConStatus
GW	3	8	EMS-Status
GW	3	8	EMS-IdleSpeed
GW	3	8	EMS-AcpeDaRos
GW	3	8	EMS-Temp
GW	3	8	EMS-Torq
TCU	1	4	TCU-Status

Node	Period (ms)	Length (byte)	Signal Representation
TCU	3	8	TCU-Temp
TCU	3	8	TCU-TurSpeed
TCU	3	8	TCU-TorqReq
EPS	1	4	EPS-Status
EPS	3	8	EPS-StrAng
CDC	1	4	CDC-Status
ASC	1	4	ASC-Status

The verification system uses a single channel for inter-node signal transmission. First, configuration rules set $T_{APO} = 1 \cdot T_{MT}$, $T_{MT} = 1\mu s$, and $C = 10$ Mbps. The communication cycle must be a common divisor of all static messages; otherwise, static messages cannot be cyclically transmitted. According to [11], the communication cycle can be set to 1 ms, with network idle time configured as 100 T_{MT} . Three different scheduling methods are used to schedule network signal data for comparative study under the same transmission rate:

- Original protocol without any optimization processing, where static timeslot length is determined by the longest message length with identical parameter configuration for all nodes, calculating static segment bandwidth utilization;
- Classical scheduling conditions where only signals with identical periods within each node can be encapsulated into one message frame [11], obtaining an encapsulation scheme and solving bandwidth utilization;
- Based on allowed timeslot multiplexing and allowing integer multiple relationship signals within the same node to be encapsulated into one message frame for transmission in one static timeslot, obtaining the corresponding encapsulation scheme and solving bandwidth utilization.

TABLE:4 Static slot parameters and bandwidth utilization under different signal encapsulation modes

Scheduling Method	T_{STS} (μs)	Bandwidth Utilization U (%)	Message Frame Count
No encapsulation	38.5	14.93	11
Classical encapsulation	38.5	23.93	8
Optimized encapsulation	38.5	29.59	8

Analysis of shows that the proposed signal encapsulation optimization method improves static segment network bandwidth utilization. The optimal message

frame set under this method is shown in . Compared with classical signal encapsulation, it improves bandwidth utilization by 5.66%, which is more favorable for future increases in automotive functional nodes and signal quantities. Since the GW-Status signal data length in the gateway is much larger than others, static timeslot parameters remain unchanged across the three scheduling methods. With more system nodes, greater signal data volume, more diverse signal period types, and more uneven signal lengths, the proposed signal encapsulation optimization will have more significant effects on bandwidth utilization improvement.

TABLE:5 Collection of message frames to be transmitted after signal encapsulation optimization

Node Name	Period (ms)	Message Frame	Signal Data Combination
ESC	4	M(1,4,1)	$S_{1,4,1} + S_{1,4,2} + S_{1,4,3} + S_{1,4,4}$
ESC	6	M(2,6,1)	$S_{2,6,1} + S_{2,6,2}$
GW	16	M(3,16,1)	$S_{3,16,1}$
GW	8	M(3,8,1)	$S_{3,8,1} + S_{3,8,2} + S_{3,8,3} + S_{3,8,4}$
GW	8	M(3,8,2)	$S_{3,8,5} + S_{3,8,6} + S_{3,8,7}$
TCU	6	M(4,6,1)	$S_{4,6,1} + S_{4,6,2}$
EPS	8	M(5,8,1)	$S_{5,8,1}$
CDC/ASC	12	M(6,12,1)	$S_{6,12,1}$

This paper employs timeslot multiplexing mechanisms to design corresponding algorithms, establishing specific scheduling positions for each signal frame in each cycle. This scheme minimizes the number of static timeslots per communication cycle and maximizes static timeslot utilization, reserving more bandwidth for future system node expansion. Moreover, this scheme allows new nodes to be added without changing system configuration, simply by configuring parameters according to the system, demonstrating excellent network scalability.

Based on completed signal encapsulation optimization, message frames are arranged in their scheduling positions within the cycle to obtain the static message schedule. Previous analysis shows the FlexRay cycle contains $NTC = 48$ communication cycles. Statistics and comparison of four methods—non-multiplexed FID with signal encapsulation optimization and multiplexed FID allocation with signal encapsulation optimization—are performed for the ratio of FIDs used for message frame transmission to total bus FIDs (static timeslot utilization within the cycle), with results shown in .

TABLE:6 Comparison of static time slot utilization under different FID allocation modes

Scheduling Method	FIDs per Communication Cycle	Static Timeslot Utilization in Cycle (%)
No encapsulation, no multiplexing	11	22.92
Classical encapsulation, no multiplexing	8	16.67
Optimized encapsulation, no multiplexing	8	16.67
Optimized encapsulation, with multiplexing	2	55.21

Analysis of shows that based on the three encapsulation methods, non-multiplexed allocation scheduling results in low static timeslot utilization within the cycle, causing many static timeslots to remain unused for transmitting any static message frames in specific cycles. Based on timeslot multiplexing mechanisms, this paper proposes corresponding algorithms to determine each message frame's position in specific cycles, minimizing FIDs per cycle and maximizing static timeslot utilization, thereby reducing the number of FIDs required by the system and facilitating future expansion. The schedule obtained by the proposed static segment message scheduling optimization algorithm is shown in .

TABLE:7 FID allocation optimization static schedule

Cycle	FID=1	FID=2	Cycle	FID=1	FID=2
1	M(1,4,1)	M(2,6,1)	24	M(1,4,1)	M(2,6,1)
2	M(3,8,1)	M(4,6,1)	25	M(3,8,1)	M(4,6,1)
3	M(3,8,2)	M(6,12,1)	26	M(3,8,2)	M(6,12,1)
4	M(5,8,1)	M(1,4,1)	28	M(1,4,1)	M(5,8,1)
5	M(6,16,1)	M(3,8,1)	35	M(5,8,1)	M(2,6,1)
10	M(3,8,2)	M(2,6,1)	36	M(1,4,1)	M(2,6,1)
11	M(5,8,1)	M(4,6,1)	37	M(6,16,1)	M(4,6,1)
12	M(1,4,1)	M(2,6,1)	41	M(3,8,1)	M(6,12,1)
16	M(1,4,1)	M(4,6,1)	42	M(3,8,2)	M(2,6,1)
18	M(3,8,1)	M(2,6,1)	43	M(5,8,1)	M(4,6,1)
19	M(5,8,1)	M(4,6,1)	44	M(1,4,1)	M(6,12,1)

Cycle	FID=1	FID=2	Cycle	FID=1	FID=2
21	M(6,16,1)	M(1,4,1)			

divides the cycle into two parts, where the first and fourth columns show communication cycle numbers, and other columns show message frames transmitted under each timeslot number in each cycle. The results demonstrate that the proposed FID optimization method effectively utilizes static segment timeslots, achieving the minimum number of FIDs per communication cycle and leaving remaining static timeslots for future network expansion without reconfiguring other nodes.

Based on the above verification and analysis, the proposed message scheduling scheme is feasible under the objectives of maximizing static segment bandwidth utilization and minimizing static timeslots per communication cycle.

4 Conclusion

For FlexRay bus static segment message scheduling, this paper proposes signal encapsulation optimization and a timeslot multiplexing-based FID allocation algorithm to determine optimal static timeslot length and corresponding static segment message schedule, effectively improving static segment bandwidth utilization and reducing FIDs required per communication cycle, providing effective guarantee for new function expansion in in-vehicle networks.

Simulation results show that the proposed signal encapsulation optimization algorithm achieves 29.59% bandwidth utilization with message frame count reduced to 8. The proposed FID allocation method reduces FIDs per communication cycle to 2 and increases static timeslot utilization within the cycle to 55.21%.

This paper's research on FlexRay static segment message scheduling optimization does not consider potential transmission faults in network communication or jitter issues within vehicles. Future research will analyze the impact of static timeslot configuration on communication reliability and real-time performance, and apply the scheduling algorithm to actual automotive systems for debugging to further optimize FlexRay message scheduling schemes.

References

- [1] Bill Chilcott. *FlexRay and its application real-time multiplexed network* [M]. UK: Wiley & Son, Ltd, Publication, 2012: 19-23.
- [2] Guo Xuejian. Modern automotive vehicle network technology application [J]. *Electronic Technology and Software Engineering*, 2014 (18): 41.
- [3] Zhang Fengdeng. *Distribution real-time systems* [M]. Beijing: Science Press, 2014.

- [4] Wei Yehua, Yan Biyun. Survey of vehicle FlexRay network scheduling algorithms [J]. *Transducer and Microsystem Technologies*, 2014, 33 (1): 5-10.
- [5] Kang M, Park K, Kim B. Determining the size of a static segment and analyzing the utilization of in-vehicle FlexRay network [C]// *Proc of International Conference on Convergence and Hybrid Information Technology*. 2008: 50-53.
- [6] Cena G, Valenzano A. On the properties of the flexible time division multiple access technique [J]. *IEEE Trans on Industrial Informatics*, 2006, 2 (2): 86-94.
- [7] Zheng Wenxue. Research on optimization design of FlexRay scheduling scheme for vehicle network [D]. Hangzhou: Zhejiang University, 2013.
- [8] Liang Jieshen, Wang Hong. Design and implementation of new total network FlexRay network communication [J]. *Microprocessors*, 2011, 32 (2): 20-23.
- [9] Wang Yuefei, Cao Sanfeng, Bi Xiang, et al. A communication mechanism of FlexRay system based on time slot dynamic allocation [J]. *Journal of Electronic Measurement and Instrument*, 2015 (2): 179-186.
- [10] Juan R O S, Kim H S. Reconfiguration of an FPGA-based time-triggered FlexRay network controller using EEDC [J]. *Journal of Circuits Systems & Computers*, 2017: 1850096.
- [11] Schmidt K, Schmidt E G. Message scheduling for the FlexRay protocol: the static segment [J]. *IEEE Trans on Vehicular Technology*, 2009, 58 (5): 2160-2169.
- [12] Kang M, Park K, Jeong M K. Frame packing for minimizing the bandwidth consumption of the FlexRay static segment [J]. *IEEE Trans on Industrial Electronics*, 2013, 60 (9): 4001-4008.
- [13] Hu You, Li Renfa, Wu Wufei. Vehicle heterogeneous network gateway data encapsulation method [J]. *Computer Engineering*, 2017, 43 (2): 1-5.
- [14] Dong Yihong, Zhao Jiayu. Research on approximation algorithm for constrained one-dimensional packing problem [J]. *Computer and Engineering and Applications*, 2003, 39 (18): 41-44.
- [15] Zhang Minghui, Han Xin. Research on one-dimensional online packing model with item size not exceeding $1/2$ [J]. *Journal of Systems Science and Mathematical Sciences*, 2017, 37 (3): 781-791.
- [16] Liu Mingming, Tong Xiaojiao, Dai Yuhong. Algorithm and latest development of packing problem [J]. *Mathematica Numerica Sinica*, 2016, 38 (3): 257-280.
- [17] Zhu Zhilin, Shi Chen, Han Jungang, et al. Packing algorithm for periodic task scheduling [J]. *Journal of Computer Applications*, 2006, 26 (3): 679-691.
- [18] Zheng F, Luo L, Zhang E. NF-based algorithms for online bin packing with buffer and bounded item size [J]. *Journal of Combinatorial Optimization*, 2013, 30 (2): 1-10.

[19] Park I, Sunwoo M. FlexRay network parameter optimization method for automotive applications [J]. *IEEE Trans on Industrial Electronics*, 2011, 58 (4): 1449-1459.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv — Machine translation. Verify with original.