

Article

Research of Message Scheduling for In-Vehicle FlexRay Network Static Segment Based on Next Fit Decreasing (NFD) Algorithm

Shiyi Jin, Mengzhuo Liu, Yujing Wu, Yihu Xu, Jingnan Zhang and Yinan Xu *

College of Engineering, YanBian University, Yanji 133002, China; 2018050378@ybu.edu.cn (S.J.); 2017050318@ybu.edu.cn (M.L.); yjwu@ybu.edu.cn (Y.W.); xuyh@ybu.edu.cn (Y.X.); 2164024212@ybu.edu.cn (J.Z.)
* Correspondence: ynzu@ybu.edu.cn; Tel.: +86-433-273-3955

Received: 26 September 2018; Accepted: 24 October 2018; Published: 26 October 2018



Abstract: FlexRay is becoming the in-vehicle communication network of the next generation. In this study, the main contents are the FlexRay network static segment scheduling algorithm and optimization strategy, improve the scheduling efficiency of vehicle network and optimize the performance of communication network. The FlexRay static segment characteristic was first analyzed, then selected bandwidth utilization as the performance metrics to scheduling problem. A signal packing method is proposed based on Next Fit Decreasing (NFD) algorithm. Then Frame ID (FID) multiplexing method was used to minimize the number of FIDs. Finally, experimental simulation by CANoe. FlexRay software, that shows the model can quickly obtain the message schedule of each node, effectively control the message payload size and reduced bus payload by 16.3%, the number of FID drops 53.8% while improving bandwidth utilization by 32.8%.

Keywords: FlexRay; static segment; signal packing; FID multiplexing; utilization

1. Introduction

With the improvement of the automobile electronization, it results in the rapid increase of modules numbers and more complicated systematic functions for realization more advanced software and electronic functions. For general examples, systems like the airbag, the self-adaptive cruise control, and the traction control are implemented on distributed electrical and electronic architecture to obtain more comfortable and safer vehicle environments.

Whereas the commonly used CAN (Controller Area Network) system has severely restricted the function expansion and cost control of vehicles [1], in 2000, FlexRay union was established by companies of BMW, Daimler Chrysler, Motorola, and Philips and the FlexRay network was developed to have a 20 M/bps of bit rate, the high throughput, and the certainty, as well as a hybrid high-speed communication network with time-division and flexible time-division multiple access. In 2010, FlexRay 3.0 protocol was published [2]. The FlexRay bus network has been applied to high-class commercial vehicles like BMW X5, BMW 7 series, and Audi A8 series [3,4]. In order to satisfy the increasing demand of communication network in vehicle electronic control system under the premise of ensuring the safety and reliability of vehicle bus network system, FlexRay message scheduling fusion technology is a one of the key points.

In this study, based on Next Fit Decreasing (NFD) algorithm for static segment of FlexRay bus, a message optimization scheme will be mentioned to achieve the high efficiency of FlexRay network. This scheme will resolve problems, like synthesizing the schedule of FlexRay static segment, solving the signal and Frame ID (FID) allocation problems, reducing the bandwidth and improving network utilization at the same time, as well as reducing the number of FID.

2. FlexRay Communication Protocol

As shown in Figure 1, the communication of FlexRay network is in a cyclical manner. Cycles on the bus are counting by cycle counters where its range is set from 0 to the maximum number of the system configuration $gCycleCountMax$. Each communication cycle consists of four parts. They are static segment (ST), dynamic segment (DS), symbol window (SW), and network idle time (NIT). Among them, static segment transmits periodic messages using time division multiple access (TDMA) technology and suitable for high-confidence and security-critical communications [5–7].

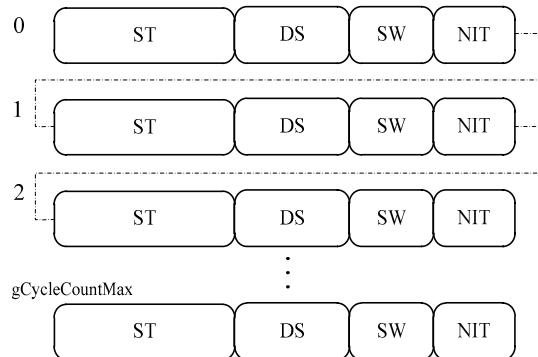


Figure 1. FlexRay communication cycle. ST, static segment; DS, dynamic segment; SW, symbol window; NIT, network idle time.

The FlexRay static segment consists of multiple time slots with the same size. The slot number is recorded by the static slot counter when the signal of frames is transmitting. A FlexRay frame is consists of the header segment, the payload segment, and trailer segment [8–10]. As shown in Figure 2, the length of valid data segment is 0–254 bytes. When a frame transmits over the bus, some bits are needed to encode it. Transmission start sequence (TSS) accounts for 3–15 bits, and frame start sequence (FSS) for 2 bits where byte start sequence (BSS) should be added at the front of every byte. Frame end sequence (FES) accounts for 2 bits. Communication idle delimiter (CID) takes another 11 bits and action point offset (APO) accounts for 1–63 macrotick (MT) [11–13].

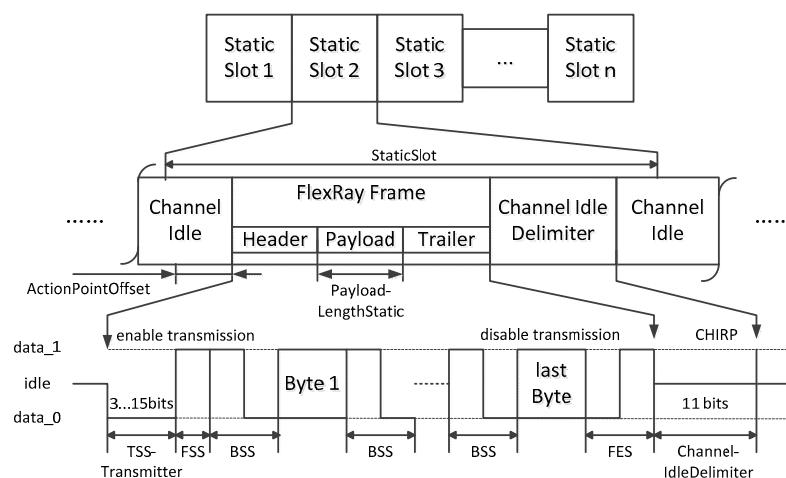


Figure 2. FlexRay message frame format.

Then, the length of frame is expressed as:

$$L_{FL}(j) = TSS + FSS + 80 + 20w_j + FES \quad (1)$$

where: L_{FL} is the length of the message frame. Eighty bits is the sum of the header segment and the trailer segment.

Static slot length can be expressed as:

$$L_{ST}(j) = 2 \times APO + \text{ceil} \left(((L_{FL} + CID) \times t_{bit} + MinPD + MaxPD)(t_{MT} \times (1 - CDM))^{-1} \right) \quad (2)$$

where L_{ST} is the length of the static time slot in MT (Macrotick); ceil is the top integral function. t_{bit} is the time to transfer each bit; $MinPD$ (Min Propagation Delay) is the minimum of propagation delay; $MaxPD$ (Max Propagation Delay) is the maximum of propagation delay; t_{MT} is the duration of each macro t ; CDM (Clock Deviation Max) is the maximum of clock deviation (0.0015).

It is assumed that there are i nodes in FlexRay bus, the FID number of each node is denote as FID_{e_i} , ($i = 1, 2, \dots, n$). The total number of FID of i nodes can be expressed as:

$$FID = \sum_{i=1}^n FID_{e_i} \quad (3)$$

Bandwidth utilization indicates how many of the allocated slots in the static segment are used to transmit signal [14]. Because of the different length of the FlexRay static message, the static slot length is determined by the length of the longest static message. Therefore, the purpose of FlexRay static segment scheduling is to ensure periodic message transmitted while minimizing the number of FIDs and maximizing bandwidth utilization.

3. Periodic Signal Packing

Having considered the possibilities of future uses like transmitting large number of signals, more efficient use of bandwidth is necessary. Hence, signal packaging becomes the key point in FlexRay network design. In general packing processes, the message signals were not distinguished from the different transmission period and were packed equally. In the previous study, large periodic signals are repeatedly transmitted because of indistinguishable periodic information from packed signals.

Assuming the previous algorithm without considering a period, packing the signals of 1 and 2 in Table 1 will have the transmission period of 5 ms and 20 ms, the corresponding repetition rates of 1 and 4, respectively. When two signals are packed and transmitted together, the smaller repetition of the two signals is selected as the repetition of the new message. So, signal 2 occupies three times more bandwidth with a data size of 8 bits in one super-period.

Table 1. Signal message of frame.

Signal	Period (ms)	Repetition	Size (bit)
1	5	1	8
2	20	4	8

In this study, the method packaging the periodic signal independently will be mentioned to avoid wasting the bandwidth and optimize the bus utilization.

Assuming that any signal on i node is assumed to be a quaternion vector $s_i = \{e_{s_i}, p_{s_i}, r_{s_i}, b_{s_i}\}$. Where, e_{s_i} is the send node of the signal s_i . p_{s_i} is the period (ms) of signal s_i , r_{s_i} is the repetition of signal s_i , b_{s_i} is the payload length(bit) of signal s_i . We packing the signal set $S = \{s_1, s_2, s_3, \dots, s_i\}$ into a message set $M = \{m_{i,1}, m_{i,2}, m_{i,3}, \dots, m_{i,j}\}$. Assume that any message on i-node to be a quaternion vector $m_{i,j} = \{e_{m_{i,j}}, p_{m_{i,j}}, r_{m_{i,j}}, b_{m_{i,j}}\}$. Where, $e_{m_{i,j}}$ is the send node message $m_{i,j}$, $p_{m_{i,j}}$ is period (ms) of message $m_{i,j}$, $r_{m_{i,j}}$ is the repetition of message $m_{i,j}$, $b_{m_{i,j}}$ is the payload length(bit) of message $m_{i,j}$. In order to realize better signals packaging, assumptions of a signal packing model are listed as below.

1. All information of signals are known including the transmission node, transmission period, signal deadline, repetition rate, and the size of signal. The value of transmission period equals to the deadline.
2. All of the transmission periods of signals on FlexRay bus are integral multiples of the communication cycle.
3. There is no error frame on FlexRay bus. Each node can transmit and receive signals.
4. Signals are only packed together when they have same ECU (Electronic Control Unit) and same repetition rates.

In the case of above assumptions, the signal packing is started as follows. Two-dimensional variables $x_{i,j}$ is pulled in and $x_{i,j} \in \{1, 0\}$. $x_{i,j} = 1$ indicates that the signal s_i is packaged into the message $m_{i,j}$ when the ECU of signal s_i and message $m_{i,j}$ is same to the transmission cycle, otherwise $x_{i,j} = 0$.

$$x_{i,j} = \begin{cases} 1, & e_{s_i} = e_{m_{i,j}} \text{ and } p_{s_i} = p_{m_{i,j}} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

In order to ensure that each signal is packaged once only, it needs a following formulation:

$$\sum_{j=1}^K x_{i,j} = 1, \quad i = 1, 2, \dots, n \quad (5)$$

The problem of signal packing can be transformed into an open packing problem. The capacity of the box can be converted to the effective payload length of the frame, and the signal is regarded as an item to be packed. Because the length of the frame's payload is indefinite, the upper and lower bound constraints are set on them. In FlexRay network, the length of the frame's effective payload segment is the signal length which is packaged into the frame. As shown in Equation (6), the conditions of the package show that the maximum payload length will be got if only the same repetition signal rates are packaged into the same message. As shown in Equation (7), we can obtain the minimum length of the payload if all the signals are in one frame in each node.

$$l^L = \max_{1 \leq i \leq n} b_{s_i} \quad (6)$$

$$l^H = \max_{1 \leq i \leq n} \sum_{r_{s_i}=1}^m b_{s_i} \quad (7)$$

It will be seen that the length of the payload of the frame is in the range of $[l^L, l^H]$. Hence, the optimal frame payload length is:

$$l^L \leq l^{best} \leq l^H \quad (8)$$

Therefore, the static segment network utilization can be expressed by Equation (9), where K is the number of packing messages.

$$\forall m_{i,j} \in M, s_i \in S \quad U = \frac{\sum_{i=1}^n s_i}{K \times \max(m_{i,j})} \quad (9)$$

The cycle communication mode is adopted of FlexRay. So, in the communication cycle, the signal of all super-periodic is retransmitted.

$$T_{sp} = LCM(p_{m_{1,1}}, p_{m_{1,2}}, p_{m_{1,3}}, \dots, p_{m_{1,j}}, p_{m_{2,1}}, p_{m_{2,2}}, p_{m_{2,3}}, \dots, p_{m_{2,j}}, \dots, p_{m_{i,j}}) \quad (10)$$

When the number of nodes and the number of signals in the FlexRay network are increased multiply, the traditional linear programming model will result in more complicated and difficult

processes, as well as longer solving times. In this study, we will use the Next Fit Decreasing (NFD) algorithm to optimize bus utilization.

First in the NFD algorithm, the signal is sorted from large to small according to the data of length, assuming that the length of the signal $s_1, s_2, s_3, \dots, s_i$ is $L_1, L_2, L_3, \dots, L_i$ respectively, the signal is packaged into a message in a given order. The signal s_1 is put into slot sl_1 firstly, if $L_1 + L_2 \leq l^{cur}$, then s_2 will put in sl_1 , otherwise it will put into sl_2 . Continually, testing whether s_2 could be put into sl_2 . If can, put into the slot sl_2 , otherwise open the slot sl_3 . Figure 3 is a signal package flow chart, the main operation flow is as follows:

- (1). Obtain the signal information corresponding to each node.
- (2). The signal of each node is arranged in ascending order of the repetition rate, and the signal of the same repetition rate in each node is arranged in descending order of the size of the data bits.
- (3). Calculate the exact value of the payload length of the frame, which means calculate the values of l^L and l^H , and initialize $l_{cur} = l^L, U = 0$.
- (4). According to the NFD algorithm, the nodes are packaged by sequentially. Finally, a message set is obtained.
- (5). Calculate a super-periodic network utilization in the case of the payload length of the current frame, that expressed as U_{cur} , if $U < U_{cur}$, update the current message set.
- (6). Set $l_{cur} = l_{cur} + 1$ and repeat steps (4) and (5), loop until $l_{cur} = l^H$.
- (7). Take the maximum value of U when the corresponding message set and its corresponding frame payload length l^{best} .

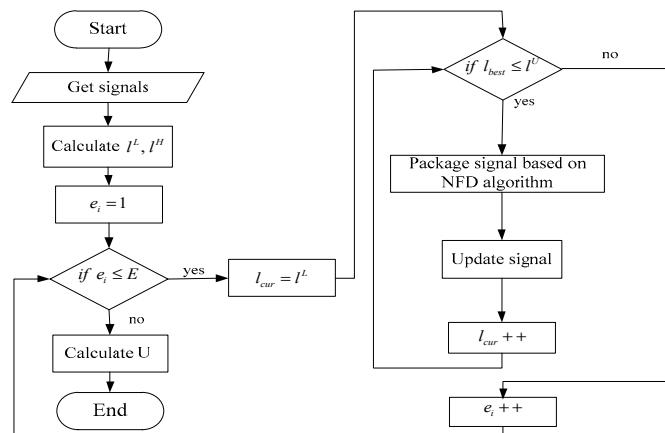


Figure 3. Frame packing flow chart.

4. Frame ID Reusing

After signal packing, we can get the best length of payload and the best network utilization. Based on this, a schedule of frame ID (FID) reusing is constructed. The reuse of FID is important to scheduling strategy, making the schedule to occupy the least slots under the requirements of real-time system, so as to retain as much free slots. As well as, when new nodes and new frames are added for scaling, not only the flexibility of the static schedule be improved, but also the network resources are saved as much as possible. Scheduling for frames of FlexRay network is shown in Figure 4.

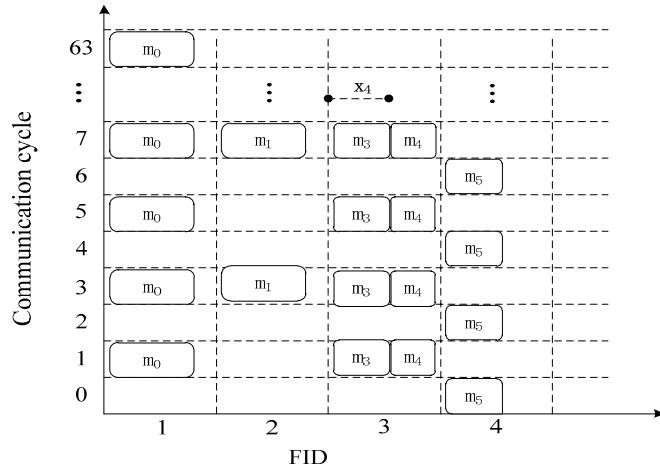


Figure 4. Frame schedule for FlexRay.

As shown in Figure 4, the repetition of message m_0, m_3, m_4, m_5 is 2, the repetition r_m of message m_1 is 4, the repetition of message is determined by sending periodic p_m and communication cycle T_{bus} .

$$r_m = \frac{p_m}{T_{bus}} \quad (11)$$

The period when message m is sent on the bus for the first time is called basic cycle, which can be represented in Figure 4 by the y-offset of the y-axis. So, the base cycle of message m_0, m_3, m_4 is 1, the base cycle of message m_1 is 3, and the base cycle of message m_5 is 0. The basic cycle constraints are as follow.

$$bc_m < r_m \quad (12)$$

The relationship between basic cycle bc_m and repetition rate r_m is shown in Equation (13), N_c is the period for sending the current message. Where, $bc_m \in \{0, 1, \dots, 63\}$, $r_m \in \{1, 2, 4, 5, 8, 10, 16, 20, 32, 40, 50, 64\}$.

$$N_c \bmod r_m = bc_m \quad (13)$$

The offset x_m is the location in the slot where message m was sent, which can be represented in Figure 4 by the x-offset of the x-axis. Such as the offset of message m_4 is 4, the range of the offset is as shown in Equation (14). Where l_s is the size of static slot, l_m is the size of message.

$$0 < x_m \leq l_s - l_m \quad (14)$$

For a set of packaged messages M , the messages are sorted according to the following rules to ensure that the messages are closely arranged in the schedule and do not intersect each other.

(1) The message having the smaller repetition rate will be sent more frequently. So, the message with a smaller repetition rate is arranged transmitting first. For the larger repetition rate of messages, it can be arranged flexibly because of they occupied fewer cycles. Thus, the remaining cycles are used to put them. Therefore, for message of each node, it should be arranged in the ascending order according to the repetition rate of packed messages.

$$\begin{aligned} \forall m \in M, a, b \in K, a \neq b, \\ r_{m_a} \leq r_{m_b}, 0 \leq a \leq b \leq k \end{aligned} \quad (15)$$

(2) For messages with the same repetition, be sure to place the larger message at first, and fill the remaining payload in that slot with the smaller message. Therefore, messages in each node with the same repetition rate should be arranged in the decreasing order of its length.

$$\begin{aligned} \forall m \in M, a, b \in K, a \neq b, \\ r_{m_a} = r_{m_b}, l_{m_a} \geq l_{m_b}, 0 \leq a \leq b \leq k \end{aligned} \quad (16)$$

(3) They should not intersect for all repeated message in the schedule arrangement. Even if messages m and m' have the same slot, the schedule does not intersect because of their different cycles.

$$\forall i, j \in N_o : (b_m + r_m \times i) \bmod N_c \neq (b_{m'} + r_{m'} \times j) \bmod N_c \quad (17)$$

All messages are sorted in ascending order by repetition rate as shown in Table 2, the repetition of m_2 is same as m_3 and m_4 , sort in descending order of payload size. When the message frame is assigned to the schedule, it is allocated from the low period. The communication cycle of FlexRay is 0–63. Because the frame is sent repeatedly according to the period, the sampling cycle T_{sp} is the super period of frames. In the case of message sent form nodes in Table 2, only 0–7 cycle needs to be considered.

Table 2. Messages required for sending of an ECU (Electronic Control Unit).

Message	Repetition	Size (bit)	Message	Repetition	Size (bit)
m_1	1	1	m_4	2	2
m_2	2	8	m_5	4	16
m_3	2	4	m_6	8	11

In order to obtain better configure of the schedule, T_{sp}/T_{bus} rows that have the same slot even though have different periods could be regarded as the same one column. Then, the reciprocal of the repetition is taken and added up. When the sum is 1, the message has been configured to fill this slot. For example, the message repetitions r_m are respectively 1, 2, 2, 2, 4, and 8 in Table 2. They have the reciprocal of 1, 1/2, 1/2, 1/2, 1/4, 1/8, respectively. If the repetition of message is 1, it indicates that one slot is full filled. If the reciprocal of m_2 and m_3 is 1/2, they will occupy one slot. In the other case, the cumulative sum of messages m_4, m_5, m_6 is 7/8. Though the sum of them is less than 1, the nodes of messages has been assigned slots, so the message m_4, m_5, m_6 will occupy a whole slot. The scheduling strategy which was provided in this paper was used in Table 2 to get the schedule shown in Figure 5.

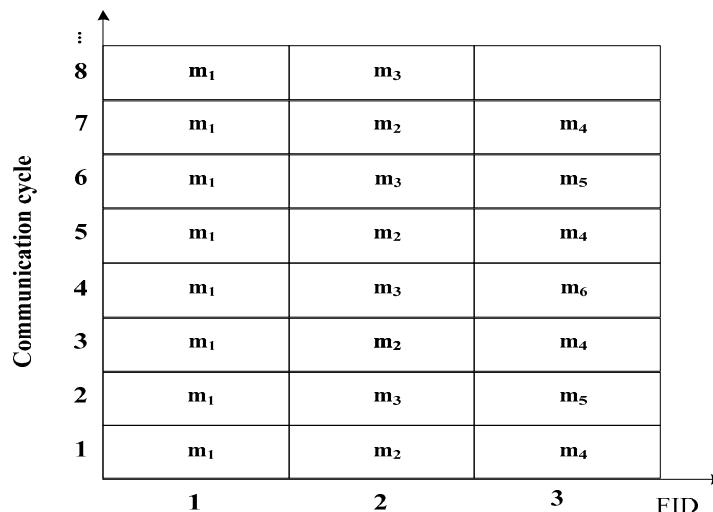


Figure 5. Message schedule for an ECU.

FID can be shared when frames are from the same node. For each node e_i , required number of *FID* is:

$$FID_{e_i} = \left[\sum_{j=1}^k \frac{1}{r_{m_{i,j}}} \right] \quad (18)$$

The equation is extended to a system with k frames for n nodes, Then, required number of *FID* is given by Equation (19).

$$FID_{e_i} = \sum_{i=1}^n FID_{e_i} \quad (19)$$

5. Experimental Results

To evaluate our message scheduling method, we applied it to the communication system which has 4 nodes in BMW X7. The duration of communication cycle is the typical value of 5 ms in where the static segment occupies 3 ms and the dynamic segment occupies 2 ms. And other simulation parameters are shown in Table 3. The table shows the configuration of FlexRay network parameters dedicated to construct the data frame. In this study, we designed experiments with two different sets of data (Tables 4 and 5). Table 4 shows the SAE benchmark data. Table 5 shows the signal parameters of BMW's 7 series where signal repetition rate is set to 1, 2, 4 and 8 respectively. Both of Tables 4 and 5 shows in detail about the period of transmitted signals of each ECU, repetition, and data length. The experiment was as follows.

Table 3. FlexRay network parameter values.

Parameter	Value	Unit	Parameter	Value	Unit
APO	1	MT	CID	11	bit
TSS	9	bit	gdBit	0.1	μs
FSS	1	bit	gdMacrotick	1	μs
BSS	2	bit	Bitrate	10	Mbit/s
FES	2	bit	t_{MT}	1	μs

APO, action point offset; TSS, transmission start sequence; FSS, frame start sequence; BSS, byte start sequence; FES, frame end sequence.

Table 4. SAE benchmark data.

Signal	ECU	Period (ms)	Repetition	Size (bit)
1		100	20	8
2		100	20	8
3	1	100	20	8
4		100	20	8
5		5	1	8
6	2	5	1	8
7		100	20	8
8	3	5	1	8
9		5	1	8
10	4	5	1	8
11		5	1	8
12	5	100	20	8
13		5	5	8
14		5	5	8
15	6	10	2	8
16		10	2	8

Table 5. FlexRay network signal set.

Signal	ECU	Repetition	Size (bit)	Signal	ECU	Repetition	Size (bit)
1	1	1	20	26		4	20
2			12	27			16
3			10	28			8
4		2	16	29			8
5			12	30	3	2	16
6			10	31			12
7			8	32			10
8			8	33			8
9		4	16	34		4	16
10			12	35			15
11			10	36			12
12			8	37			8
13			8	38		8	16
14		8	20	39			10
15			16	40			8
16			12	41			8
17			8	42	4	2	16
18	2	1	12	43			12
19			10	44			10
20			2	45			8
21		2	20	46			8
22			16	47		8	10
23			10	48			8
24			8	49			8
25			2	50			2

(1) Using the SAE benchmark data. Let signals of Table 4 as experimental raw data. Repetition values are set to 1, 2, and 20. The sample signal's length is equal to 8 bits and randomly expanded. The proposed algorithm and BCBFD algorithm which BC and BFD is an abbreviation of Bandwidth Consumption and Best Fit Decreasing respectively [15], are used to the signal set packing, respectively. Compared the upper and lower limits of payload data payload length, as well as bus utilization and the number of slots.

Figure 6 shows that how the upper and lower bound of the frame's payload of l^L , l^H change when the number of signals increases. As shown in Figure 6, the lower bound of the frame payload length of the two algorithms is fixed and have equal values. It is determined by the maximum value of the length of all signals in the node. It results in the effective length of 8 bits in a single signal payload without considering a signal expanding. Even in the case of continually increased l^H , it can be seen that the NFD scheduling algorithm can effectively control the increase of the upper bound of the payload, and save operation time.

As shown in Figure 7, the result of the curve of network utilization indicates that the average network utilization rate of scheduling algorithm proposed in this paper has a higher value than the average network utilization rate of BCBFD algorithm. Even though the network utilization of BCBFD algorithm increases with the number of signals, there is a gap that compared with the algorithm in this paper. This shows that the NFD scheduling algorithm has a stronger ability to save network resources.

The comparison of the slot number is shown in Figure 8. The number of FIDs in BCBFD algorithm increases linearly with the number of messages, while the number of FIDs in NFD algorithm increases slowly. The conclusion is that the NFD scheduling algorithm is stronger than the BCBFD algorithm in resolving scheduling problems of static segment.

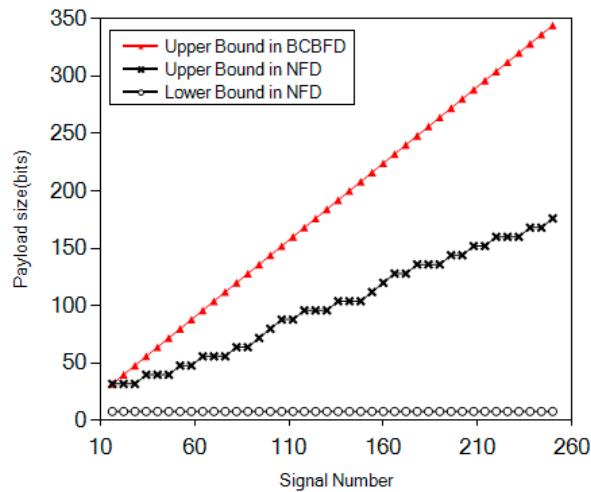


Figure 6. Bound limits of the frame payload segment.

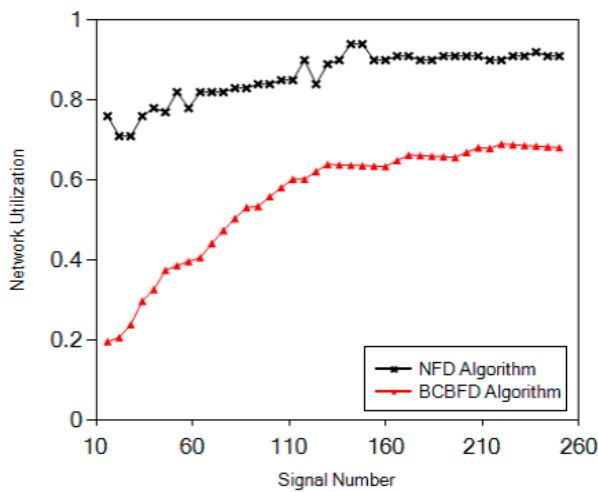


Figure 7. Comparison of network utilization.

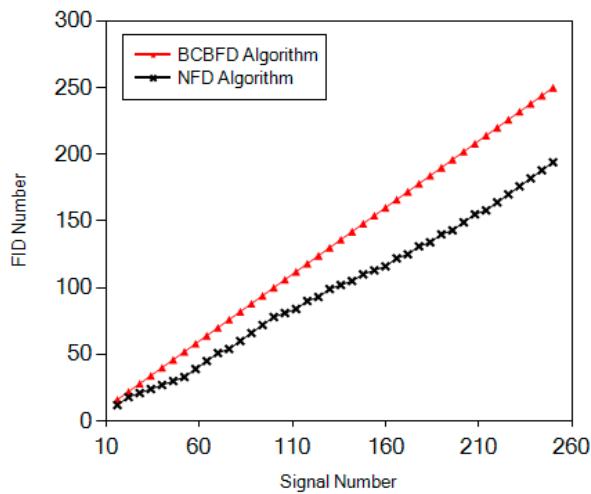
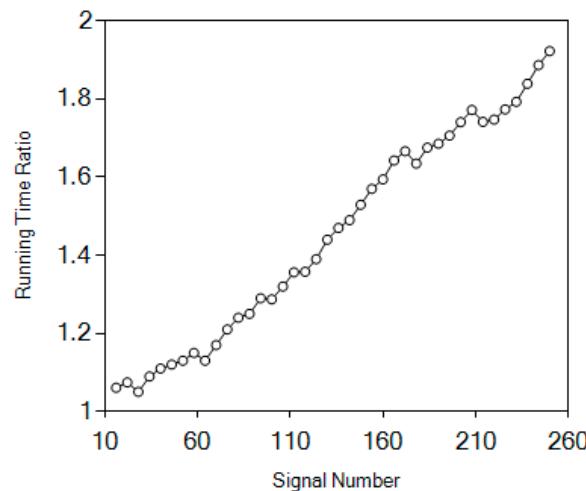


Figure 8. Comparison of frame ID (FID).

BCBFD algorithm and NFD scheduling algorithm execution time ratio is shown in Figure 9. The bound of upper and lower of the length payload segment are precisely calculated, also effectively reduced. The number of iterations of the algorithm is reduced, and the execution efficiency of the algorithm is improved.

**Figure 9.** Running time ratio.

(2) Using the BMW's 7 series data, to show the benefits of the NFD algorithm, we used the illustrative case, and compared with BCBFD algorithm [15]. The experiment is based on the vehicle bus simulation tools Network Designer and CANoe. FlexRay where the network system is built on. The case uses the data of Table 5. The signal data length is 2–20 bit, the payload length is taken as 20–56 bits, and the repetition rate of BMW 7 Series are 1, 2, 4, and 8.

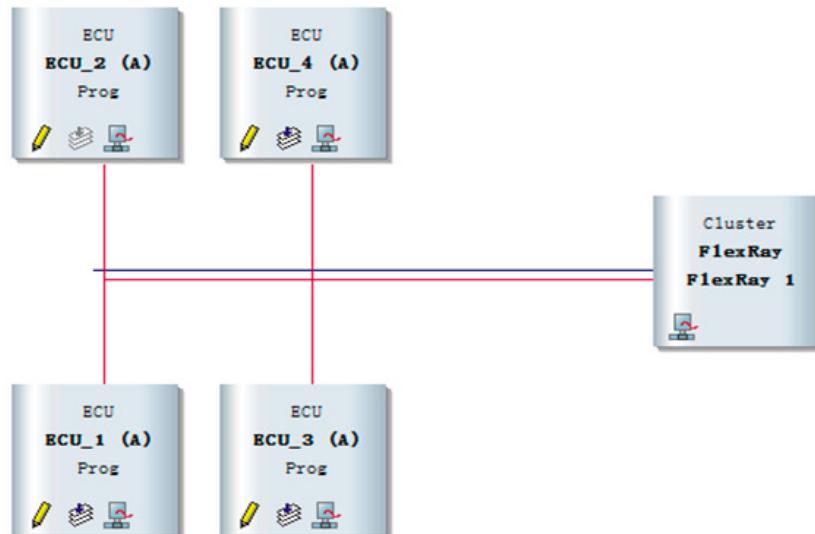
Using the data in Table 5, two algorithms were used to packing the signal sets into frames to obtain the respective package combinations. They are shown in Tables 6 and 7. Without modification in the numbers of both ECU and transmitted signal, based on the NFD algorithm proposed in this paper and the BCBFD algorithm [15], a reconstruction of transmitted signal is executed. Then, the database is built by Network Designer, and imported into the softwares CANoe and FlexRay. As shown in Figure 10, four node ECUs are set up in the simulation network of CANoe. FelxRay that are used to transmitted frames. The proposed NFD algorithm and BCBFD algorithm were simulated and compared. Network performances of two different algorithms are compared and shown in Table 8. Data in the table shows that the busload is reduced by 16.3%, the bandwidth utilization is increased by 32.8%, and the number of frame FIDs is reduced by 53.8%. The proposed algorithm can reduce the number of time slots and effectively reduce the network payload, and the network utilization has also been improved.

Table 6. Signal packing for Next Fit Decreasing (NFD) scheduling.

Message	Signal	ECU	Repetition	Size (bit)
1	1, 2, 3	1	1	42
2	4, 5, 6, 7, 8		2	54
3	9, 10, 11, 12, 13		4	54
4	14, 15, 16, 17		8	56
5	18, 19, 20	2	1	24
6	21, 22, 23, 24, 25		2	56
7	26, 27, 28, 29		4	52
8	30, 31, 32, 33	3	2	46
9	34, 35, 36, 37		4	51
10	38, 39, 40, 41		8	42
11	42, 43, 44, 45, 46	4	2	54
12	47, 48, 49, 50		8	28

Table 7. Signal packing for BCBFD (Bandwidth Consumption and Best Fit Decreasing) scheduling.

Message	Signal	ECU	Repetition	Size (bit)
1	1, 14, 4	1	1	40
2	9, 15, 2		1	48
3	5, 10, 16, 3		2	48
4	6, 11, 7, 8, 12		2	46
5	13, 17		4	24
6	21, 26, 22	2	2	40
7	27, 18, 19		1	44
8	23, 24, 28, 29, 20, 25		2	48
9	30, 34, 38, 35	3	2	48
10	31, 36, 32		2	49
11	33, 39, 37, 40, 41		2	42
12	42, 43, 44, 47, 45	4	2	48
13	46, 48, 49, 50		2	34

**Figure 10.** The node set of FlexRay simulation system.**Table 8.** Network performance comparison.

Network Performance	NFD Algorithm	BCBFD Algorithm
Total frames	1025	1225
Null frames	0	0
Busload	7.23%	8.75%
Bandwidth utilization	83%	62.5%
FID number	6	13

6. Conclusions

This study focused on the FlexRay static segment scheduling, and analyzed transmission characteristics of static segment message, then, proposed the network evaluation criteria. Using NFD algorithm, the signal encapsulation problem is converted to a packing problem. At the same time, a scheduling strategy for message *FID* multiplexing is established to optimize the transmission of message. The simulation system was established by Network Designer, FlexRay software and CANoe. FlexRay software. The simulation results show that the proposed algorithm could reduce the bus load by 16.3%, the bandwidth utilization by 32.8%, as well as decrease the number of FIDs by 53.8%. The FlexRay static segment message improves the transmission efficiency.

Author Contributions: Y.X. (Yinan Xu): conceptualization and supervision; S.J.: methodology and writing of original draft; M.L. and J.Z.: software and formal analysis; Y.W. and Y.X. (Yihu Xu): validation and review & editing of original draft.

Funding: This research was funded by National Natural Science Foundation of China Grant Number 61763047.

Acknowledgments: This research was supported by National Natural Science Foundation of China Grant Number 61763047.

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

References

1. Dvorak, J.; Hanzalek, Z. Using Two Independent Channels with Gateway for FlexRay Static Segment Scheduling. *IEEE Trans. Ind. Inform.* **2016**, *12*, 1887–1895. [[CrossRef](#)]
2. FlexRay Consortium. FlexRay Communications System Protocol Specification Version 3.0.1. Available online: www.flexray.com (accessed on 28 March 2017).
3. Kötz, J.; Poledna, S. Making FlexRay a reality in a premium car. In Proceedings of the SAE Convergence 2008, Detroit, MI, USA, 20–22 October 2008.
4. Cakmak, C.; Schmidt, E.G.; Schmidt, K.W. A Fast and Optimal Static Segment Scheduling Method for FlexRay v3.0. In Proceedings of the 25th Signal Processing and Communications Applications Conference, Antalya, Turkey, 15–18 May 2017. [[CrossRef](#)]
5. Sagsteller, F.; Lukasiewycz, M.; Chakraborty, S. Generalized Asynchronous Time-Triggered Scheduling for FlexRay. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2017**, *36*, 214–226. [[CrossRef](#)]
6. Schmidt, K.; Schmidt, E.G. Message Scheduling for the FlexRay Protocol: The Static Segment. *IEEE Trans. Technol.* **2009**, *58*, 2170–2179. [[CrossRef](#)]
7. Souto, P.F.; Portugal, P.; Vasques, F. Reliability Evaluation of Broadcast Protocols for FlexRay. *IEEE Trans. Veh. Technol.* **2016**, *65*, 525–541. [[CrossRef](#)]
8. Xu, Y.N.; Kim, Y.E.; Cho, K.J.; Chung, J.G.; Huang, X.M. Implementation of FlexRay CC and BG Protocols with Application to a Robot System. *Int. J. Comput. Aided Eng. Technol.* **2010**, *2*, 232–237. [[CrossRef](#)]
9. Shreejith, S.; Fahmy, S.A. Extensible FlexRay Communication Controller for FPGA-Based Automotive Systems. *IEEE Trans. Veh. Technol.* **2015**, *64*, 453–465. [[CrossRef](#)]
10. Xu, Y.N.; Kim, Y.E.; Cho, K.J.; Chung, J.G.; Lim, M.S. Implementation of FlexRay Protocol with an Automotive Application. In Proceedings of the IEEE International SoC Design Conference, Busan, Korea, 24–25 November 2008. [[CrossRef](#)]
11. Hua, Y.; Liu, X.; He, W.B.; Feng, D. Design and Implementation of Holistic Scheduling and Efficient Storage for FlexRay. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 2529–2539. [[CrossRef](#)]
12. FlexRay Consortium. FlexRay Communications System Protocol Specification Version 2.1. Available online: www.flexray.com (accessed on 5 June 2006).
13. Zhao, R.; Qin, G.H.; Liu, J.Q. Optimal Scheduling of the Flexray Static Segment Based on Two-Dimensional Bin-Packing Algorithm. *Int. J. Automot. Technol.* **2016**, *17*, 703–715. [[CrossRef](#)]
14. Park, I.; Sunwoo, M. FlexRay Network Parameter Optimization Method for Automotive Applications. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1449–1458. [[CrossRef](#)]
15. Kang, M.; Park, K.; Jeong, M. Frame packing for minimizing the bandwidth consumption of the FlexRay static segment. *IEEE Trans. Ind. Electron.* **2013**, *60*, 4001–4007. [[CrossRef](#)]



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).