REAL TIME SCHEDULING AND ANALYSIS FOR CAN MESSAGES WITH OFFSETS

Yang Chen
ABSTRACT

The Controller Area Network (CAN) was designed as a simple, efficient and robust broadcast communications bus for in-vehicle networks; and it has been widely used by automotive manufacturers in recent decades. In the development of CAN system, schedulability analysis theory plays an important role in evaluating whether the system is schedulable or not. This is accomplished by calculating the WCRT (worst-case response time) of every message, and comparing it with the corresponding deadline. This theory has been recognized by major automotive manufacturers; and has been applied to the design of the in-vehicle networks of a wide range of cars since 1998.

In recent years, the sophistication and complexity of in-vehicle communications has seen a notable increase in the number of transmitted messages, thus decreasing the schedulability of the networks. In this context, scheduling messages with offsets has emerged as an effective method to improve the schedulability of modern CAN networks. However, it remains unsolved how to assign offsets to the messages in order to obtain a better schedule with respect to the WCRT; and how to analyze the schedulability of messages with offsets, when they are sent using FIFO (first-in-first-out) queues. These issues limit the practical application of offset-based message scheduling.

To tackle these issues, in this dissertation, an offset assignment method; a schedulability analysis method for CAN messages with offsets in a system using FIFO queues; and a schedulability comparison between priority queues and FIFO queues for CAN messages
with offsets are proposed. The contributions of this dissertation are as follows.

- An offset assignment method that employs the Simulated Annealing (SA) algorithm for a better offset assignment with respect to the WCRT. The method initializes offsets by restricting the objective function of SA, which is selected as Maximum Interference Function obtained from messages in each station, as small as possible. Then, according to the WCRT corresponding to the initial offsets, the method modifies the objective function and reforms the offsets to guarantee that all messages meet their deadlines. Compared to previous work, the proposed method improves the schedulability of CAN messages effectively.

- A schedulability analysis method for CAN messages with offsets in a system using FIFO queues. In the analysis method, a new critical instant theorem is proposed to locate the worst case situation for a given message. Then, based on this theorem two algorithms for calculating the WCRT are proposed. The new critical instant theorem is proved to be valid; and several experiments, using synthetically generated message sets and a real-world message set provided by an automobile manufacturer, are conducted to confirm the method to be effective.

- A schedulability comparison between priority queues and FIFO queues for CAN messages with offsets. The results show that priority queues achieve higher schedulability for normal message sets, but FIFO queues are better for some special message sets. To combine the advantages of the two types of queues, a new scheduling method that uses both priority and FIFO (P&F) queues in a single station is proposed. The schedulability analysis results using a real-world message set — provided by an automobile manufacturer — show that the combination of priority and FIFO queues can achieve higher schedulability than either priority or FIFO queues alone.
CONTENTS

ABSTRACT

1 INTRODUCTION
  1.1 Backgrounds ............................................. 1
  1.2 Overview of This Dissertation .......................... 3
  1.3 Organization ............................................. 4

2 SCHEDULABILITY ANALYSIS THEORY OF CAN MESSAGES  5
  2.1 Overview .................................................. 5
  2.2 The Controller Area Network ............................ 6
    2.2.1 Protocol ............................................. 6
    2.2.2 Messages Queueing ................................. 7
    2.2.3 Priority Based Arbitration ......................... 7
    2.2.4 Bit Stuffing ....................................... 9
  2.3 Schedulability Analysis for CAN Messages ............ 10
    2.3.1 Modeling ............................................ 10
    2.3.2 The Analysis Method ............................... 11
    2.3.3 Development ....................................... 14
  2.4 Effects of Offset ....................................... 14
  2.5 Schedulability Analysis for CAN Messages with Offsets . 16
<table>
<thead>
<tr>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 AN OFFSET ASSIGNMENT METHOD FOR IMPROVING SCHEDULABILITY</td>
</tr>
<tr>
<td>1.1 Overview</td>
</tr>
<tr>
<td>1.2 Terminology and notation</td>
</tr>
<tr>
<td>1.3 Offset Initialization</td>
</tr>
<tr>
<td>1.3.1 Simulated Annealing</td>
</tr>
<tr>
<td>1.3.2 The Proposed Offset Initialization Algorithm</td>
</tr>
<tr>
<td>1.4 Offset Reformation</td>
</tr>
<tr>
<td>1.5 Evaluation</td>
</tr>
<tr>
<td>1.5.1 Experiments on Message Sets Provided by Automakers</td>
</tr>
<tr>
<td>1.5.2 Experiments on Message Sets Generated by NETCARBENCH</td>
</tr>
<tr>
<td>1.6 Summary</td>
</tr>
<tr>
<td>2 WCRITICAL INSTANT IN THE OFFSET MODEL</td>
</tr>
<tr>
<td>2.1 Critical Instant in The Offset Model</td>
</tr>
<tr>
<td>2.2 IF&amp;MIF</td>
</tr>
<tr>
<td>2.3 WCRT Calculation</td>
</tr>
<tr>
<td>3 AN OFFSET ASSIGNMENT METHOD FOR IMPROVING SCHEDULABILITY</td>
</tr>
<tr>
<td>3.1 Overview</td>
</tr>
<tr>
<td>3.2 Terminology and Notation</td>
</tr>
<tr>
<td>3.3 Offset Initialization</td>
</tr>
<tr>
<td>3.3.1 Simulated Annealing</td>
</tr>
<tr>
<td>3.3.2 The Proposed Offset Initialization Algorithm</td>
</tr>
<tr>
<td>3.4 Offset Reformation</td>
</tr>
<tr>
<td>3.5 Evaluation</td>
</tr>
<tr>
<td>3.5.1 Experiments on Message Sets Provided by Automakers</td>
</tr>
<tr>
<td>3.5.2 Experiments on Message Sets Generated by NETCARBENCH</td>
</tr>
<tr>
<td>3.6 Summary</td>
</tr>
<tr>
<td>4 WCRITICAL INSTANT IN THE OFFSET MODEL</td>
</tr>
<tr>
<td>4.1 Overview</td>
</tr>
<tr>
<td>4.2 System model, terminology and notation</td>
</tr>
<tr>
<td>4.2.1 Modeling</td>
</tr>
<tr>
<td>4.2.2 Assumptions</td>
</tr>
<tr>
<td>4.3 The Worst Case of Messages in FIFO Queues</td>
</tr>
<tr>
<td>4.3.1 Definitions of Critical Instant and Critical Instant Candidate</td>
</tr>
<tr>
<td>4.3.2 Critical Instant Theorem</td>
</tr>
<tr>
<td>4.4 The Proposed algorithms for WCRT calculation</td>
</tr>
<tr>
<td>4.4.1 The Exact Algorithm</td>
</tr>
<tr>
<td>4.4.2 The Approximate Algorithm</td>
</tr>
</tbody>
</table>
4.4.3 Computational Complexity Analysis ........................................... 52

4.5 Evaluation ...................................................................................... 52
  4.5.1 Experiments on NETCARBENCH-generated Message Sets ........ 53
  4.5.2 Experiments on An Automaker-provided Message Set .......... 54
  4.5.3 Experiments about Effectiveness of Offset Assignment ........ 55

4.6 Extension with Consideration of Jitter .............................................. 55
  4.6.1 Effects of Jitter on the Proposed Method ......................... 55
  4.6.2 Extended Successive Affect Frame Sequences ................. 56
  4.6.3 Worst-case Jitter Occurred $CIC_k$ of $\Gamma_{A(i,m)}^k$ .......... 56

4.7 Summary ......................................................................................... 57

5 SCHEDULABILITY COMPARISON BETWEEN PRIORITY QUEUES AND FIFO QUEUES
FOR CAN MESSAGES WITH OFFSETS .................................................. 59
  5.1 Overview ......................................................................................... 59
  5.2 Schedulability Comparison .............................................................. 60
    5.2.1 Experiments on Normal Message Sets ............................. 60
    5.2.2 Experiments on Special Message Sets ............................. 62
  5.3 Scheduling Messages with Both FIFO and Priority Queues ............... 63
  5.4 Evaluation of the P&F scheduling method ................................. 65
  5.5 Summary ......................................................................................... 66

6 CONCLUSIONS AND FUTURE WORKS .............................................. 67
  6.1 Conclusions ......................................................................................... 67
  6.2 Future Works ...................................................................................... 69

ACKNOWLEDGMENT ............................................................................. 75

LIST OF PUBLICATIONS BY THE AUTHOR ........................................... 77
LIST OF FIGURES

2.1 CAN bus architecture .................................................. 6
2.2 Standard format data frame of a CAN message ................. 7
2.3 Messages queuing ......................................................... 8
2.4 Properties of message $\tau_i$ ........................................ 15
2.5 Effects of offset ......................................................... 16
2.6 Examples of $IF$ and $MIF$ ........................................... 19
2.7 An example of saturation addition and $EIT$ .................... 20
3.1 Comparison of $MIF$s in messages with different offsets. .... 27
4.1 Properties of message $\tau_i$ for FIFO-queue based analysis .... 37
4.2 Effects of the earlier queued frame ................................. 39
4.3 An example of a critical instant candidate ........................ 41
4.4 Situation $S_1$ ............................................................. 43
4.5 From situation $S_1$ to $S_2$ ........................................... 44
4.6 From situation $S_2$ to $S_3$ ........................................... 45
4.7 An example of using $IF$ to calculate response time of $\Gamma_{A(0,1)}$ in its $CIC$ .... 48
4.8 An example of using $MIF$ to calculate $R_{A(0,1)}$ ............... 51
4.9 An example showing effects of jitter on the queuing order of messages .... 56
5.1 WCRT and WCRR results of message sets generated by NETCARBENCH. The two message sets are configured as a typical 500kbps powertrain network with 31.3% bus load, 42 messages and 54.8% bus load, 58 messages. 61

5.2 WCRR results of message sets generated by NETCARBENCH. Each station is configured to only include messages with the same period. The properties of the two message sets are 52.5% bus load, 79 messages, and 79.9% bus load, 92 messages. 63

5.3 The P&F station 64

5.4 WCRR results of the message set provided by the automobile manufacturer. The message set has 53.3% bus load and 65 messages. All message properties, including the offset, were configured by the manufacturer. 65
### List of Tables

3.1 Comparison of the WCRTs in messages with different offsets ............... 26
3.2 Experiment 3.1: The message set is provided by an automaker for a system
with 15 stations, 65 messages and 35% bus load .......................... 30
3.3 Experiment 3.2: The message set is provided by another automaker for a
system with 14 stations, 68 messages and 65% bus load .................. 31
3.4 Experiment 3.3: The message set is generated by NETCARBENCH for a
system with 5 stations, 75 messages and 55% bus load .................... 31
3.5 Experiment 3.4: The message set is generated by NETCARBENCH for a
system with 5 stations, 74 messages and 65% bus load .................... 32
4.1 A message set example ................................................................... 41
4.2 Experiment 4.1: Error and run time comparisons between the exact and
approximate algorithms based on 10 message sets generated by NETCAR-
BENCH ............................................................... 54
4.3 Experiment 4.2: Error and run time comparisons between the exact and
approximate algorithms based on a real message set ....................... 54
4.4 Experiment 4.3: A WCRT comparison of a real message set with and with-
out offset ............................................................. 55
CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

In recent years, the electronic control systems of automobiles have become much more sophisticated and complex. A modern automobile may have as many as 70 electronic control units (ECUs) for various subsystems. Although some of these from independent subsystems, others are connected by the in-vehicle network, and communications among them are essential.

The Controller Area Network (CAN) is a serial communication bus designed to provide simple, efficient and robust communications for in-vehicle networks. CAN was developed by Robert Bosch GmbH in the beginning of 1983 and later presented to the Society of Automotive Engineers (SAE) Congress in 1986. In 1987 the first CAN controller chips were released by Intel (82526) and Philips (82C200). In the early 1990s Bosch submitted the CAN specification [1] for standardisation, leading to publication of the first ISO standard for CAN in 1993 [2].

In an early stage, a common misconception was that although the protocol was suitable for guaranteeing the latency of the message with the highest priority, it was not able to guar-
antee the latency of messages with lower priority. That misconception affected negatively the popularization of CAN until Tindell et al. presented their schedulability analysis theory on CAN messages[3, 4, 5]. This theory consists of calculating the WCRT of each message, and comparing it with the corresponding deadline to determine whether the network is schedulable or not.

Tindell’s work was applied by Volvo Car Corporation in the configuration and analysis of the CAN buses for the Volvo S80 [6]. Following the success of this project, Tindell’s analysis is used as the basis of a commercial CAN schedulability analysis tool by Volcano Communications Technologies AB, which is then employed by a large number of automotive manufacturers for the design of their in-vehicle networks since 1998. Prior to Tindell’s work, CAN was typically used with bus load around 30 or 40% in automotive applications [10]. The importance of Tindell’s schedulability analysis is also shown by the fact that it has led to a large body of literature trying to improve or complete the original proposal.

With the development of the automotive industry, new innovative features are being introduced by the automotive manufacturers to distinguish their cars from the competition; or to satisfy new legal regulation. For this reason, the number of transmitted messages in the networks has rapidly increased, and therefore their schedulability has decreased. An effective method for improving the schedulability of networks is to schedule messages with appropriate offsets to decrease their WCRT[7, 8]. However, it remains unsolved how to assign offsets to each message in order to obtain a better schedule with respect to WCRT; and how to analyze the schedulability of messages with offsets, when they are sent using FIFO (first-in-first-out) queues. These issues limit the practical application of offset-based message scheduling.
1.2 OVERVIEW OF THIS DISSERTATION

For the purpose of increasing the utility of the offset-based message scheduling, so that to improve the schedulability for the modern CAN networks, an offset assignment method; a schedulability analysis method for CAN messages with offsets in a system using FIFO queues; and a schedulability comparison between priority queues and FIFO queues for CAN messages with offsets are proposed in this dissertation.

The offset assignment method employs the Simulated Annealing (SA) algorithm for a better offset assignment with respect to the WCRT. The method initializes offsets by restricting the objective function of SA, which is selected so that Maximum Interference Function obtained from messages in each station, is as small as possible. Then, according to the WCRT corresponding to the initial offsets, the method modifies the objective function and reforms the offsets to guarantee that all messages meet their deadlines. To evaluate the proposed method, experiments are conducted to compare the schedulability between the proposed method and the previous method [8]. New terms — Deadline Rate and Worst-case Response Time Rate — are proposed for the schedulability comparison (See Section 3.2). The results show that the proposed method can decrease the average Worst-case Response Time Rate by 4% to 33%, maximum Worst-case Response Time Rate by 3% to 35%, compared to the previous method. It confirms that the proposed method can effectively improve the schedulability of CAN messages.

Regarding to the schedulability analysis method for CAN messages with offsets in a system using FIFO queues, a new critical instant theorem is proposed to locate the worst case situation for a given message. Then, based on this theorem two algorithms for calculating the WCRT are proposed — an exact algorithm for accurate calculation; and an approximate algorithm for rapid estimation. The exact algorithm is computationally demanding, and is suitable for evaluating the results of the approximate algorithm or for cal-
culations in small systems. In contrast, the approximate algorithm can estimate the WCRT with limited errors while having a low computational complexity; and it is useful for larger systems. It is proved that the exact algorithm is accurate and the approximate algorithm is sufficient safety for the WCRT calculation. Furthermore, experiments are conducted to validate the above two algorithms by using synthetically generated message sets and a real-world message set from an automobile manufacturer. The results confirm the effectiveness of the proposed method.

Finally, a schedulability comparison between priority queues and FIFO queues for CAN messages with offset is conducted. The Deadline Rate and Worst-case Response Time Rate are also used for the schedulability comparison of the different queues (See Section 5.2). The results show that priority queues achieve better results for normal message sets, but FIFO queues are better for some special message sets. To combine the advantages of the two types of queues, a new P&F scheduling method, which uses both priority and FIFO queues in a single station, is proposed. The schedulability analysis results on a real-world message set from an automobile manufacturer show that the combination of priority and FIFO queues can achieve higher schedulability than either priority or FIFO queues alone.

1.3 ORGANIZATION

The remainder of this dissertation is organized as follows. Chapter 2 reviews the CAN bus protocol, existing schedulability analysis theory, and the approach for scheduling messages with offsets. After that, Chapters 3, 4 and 5 describes the three main contributions. Finally, Chapter 6 concludes this dissertation and discusses future works.
CHAPTER 2

SCHEDULABILITY ANALYSIS THEORY
OF CAN MESSAGES

2.1 OVERVIEW

This chapter reviews the main concepts of CAN and its existing analysis methods. Section 2.2 represents the elements of the CAN protocol and the system model on which the schedulability analysis is based. Section 2.3 explains the schedulability analysis for CAN messages, followed by a briefly review of related works. The definition and effects of offset are introduced in Section 2.4. Finally, an existing approach to schedulability analysis of messages with offsets is explained in Section 2.5.
2.2 THE CONTROLLER AREA NETWORK

The Controller Area Network (CAN) was designed as a simple and robust broadcast bus where a number of station are connected to the bus via an interface and operate at speeds of up to 1 Mbit/s. A simple CAN bus architecture is shown in Figure 2.1.

2.2.1 PROTOCOL

Message transfer over CAN is controlled by 4 different types of frame: Data frames, Remote Transmit Request (RTR) frames, Overload frames and Error frames. The layout of a standard format data frame is shown in Figure 2.2. Each CAN data frame is required to have a unique identifier. Identifiers may be 11-bit (standard format) or 29-bit (extended format). For transmitters, the identifier is used as a priority value to determine which message among those contending for the bus will be transmitted next. Receivers can use the identifier to filter out messages that they are not interested in, to reduce the associated processing load on the receiver’s host microprocessor.
2.2.2 MESSAGES QUEUEING

Messages are queued in the stations before being transmitted to the CAN bus. The queues are memory implemented as dual ports and shared between the host processor and the CAN controller. An example is given in Figure 2.3. In this example, the host processor queues the ID:1 message to the queue, where ID:2 and ID:3 messages already exist. As can be seen, after queuing the ID:1 message, while the CAN controller using the priority queue will attempt to transmit the highest-priority message (ID:1) to the CAN bus first, the CAN controller using the FIFO queue will attempt to transmit the first queued message (ID:2) to the CAN bus first.

2.2.3 PRIORITY BASED ARBITRATION

The CAN physical layer supports two states termed dominant ('0') recessive ('1'). If two or more CAN controllers are transmitting at the same time and at least one of them transmits a '0' then the value on the bus will be a '0'. This mechanism is used to control access to the bus and also to signal errors.

In the CAN protocol, stations wait until the bus idle period is detected before attempting to transmit. When two or more stations start transmitting at the same instant, they can monitor each bit on the bus to determine who should transmit the message with highest priority (i.e., the one with a numerically lower identifier). This allows deciding whether
the message transmission should be continued or delayed until the bus becomes idle again. As the message identifiers are unique, a station transmitting the last bit of the identifier field, without detecting a '0' bit that it did not transmit, must be transmitting the message with the lowest numerical value and hence the highest priority that was ready at the start of arbitration. This station then continues to transmit the remainder of its message, all other stations having backed off.

The requirement for a station to be able to overwrite a recessive bit, and the transmitting station detect this change, limits the combination of physical length and speed of CAN bus. The duration of each bit must be sufficient for the signal to propagate the length of the
network. This limits the maximum data rate to 1Mbit/s for a network up to 40m in length or to 125Kbit/s for a 500m long network.

The arbitration mechanism employed by CAN results in messages being sent as if all of the stations shared a single global priority-based queue. In effect, messages are sent on the bus according to fixed priority non-preemptive scheduling.

The above high level description is a somewhat simplified view of the timing behaviour of CAN. CAN does not have a global concept of time, rather each CAN controller typically has its own clock which may drift with respect to the clocks of other stations. The CAN protocol therefore requires that stations re-synchronise on each message transmission. Specifically, every station must synchronise to the leading edge of the start of frame bit caused by whichever station starts to transmit first.

2.2.4 BIT STUFFING

As the bit patterns '000000' and '111111' are used to signal errors, it is essential that these bit patterns are avoided in the variable part of a transmitted message. The CAN protocol therefore requires that a bit of the opposite polarity is inserted by the transmitter whenever 5 bits of the same polarity are transmitted. This process is referred to as bit-stuffing, and is reversed by the receiver.

Stuff bits increase the maximum transmission time of CAN messages. In worst case, each stuff bit begins a sequence of 5 bits that is itself subject to bit stuffing. Including stuff bits and the inter-frame space, the maximum transmission time $C_i$, of a CAN message $\tau_i$ containing data bytes is given by:

$$C_i = \left( \left\lceil \frac{g + 8s_i - 1}{4} \right\rceil + g + 8s_i + 13 \right) t_{bit}$$

(2.1)

where $g = 34$ for standard CAN (11-bit identifiers) or $g = 54$ for extended CAN (29-bit identifiers).
identifiers). $t_{\text{bit}}$ denotes the time it takes to transmit one bit. $s_i$ denotes the number of bytes in the data field of the frame.

2.3 SCHEDULABILITY ANALYSIS FOR CAN MESSAGES

The schedulability analysis for CAN builds on previous research into fixed priority scheduling of tasks on single processor systems. In 1994 Tindell et al. [3, 4, 5] showed how research into fixed priority preemptive scheduling for single processor systems could be adapted and applied to the scheduling of messages on CAN.

In Tindell’s works, a method of calculating the worst-case response times of CAN message was proposed. Using this method it became possible to analyze CAN based systems for timing correctness, providing guarantees that all messages and the signals that they carry would meet their deadlines.

In 2007, Davis et al. [9] found a significant flaw in the schedulability analysis given by Tindell. The flaw could potentially result in the original analysis providing guarantees for messages which could in fact miss their deadlines during network operation. A paper [10] containing a review of Tindell’s work was presented, which can be seen as the current edition of schedulability analysis for CAN.

The revised schedulability analysis for CAN based on [10] is introduced in this section. First, the system model for the analysis of CAN is explained. Then, the WCRT calculation method is introduced and followed by a review of other related works showing development in the schedulability analysis theory.

2.3.1 MODELING

The system for analysis is denoted by $\Theta$ that consists of a CAN bus and multiple CAN stations. Each station of $\Theta$ is denoted as $U_I$, where $I \in \mathbb{Z}^+$. For each $U_I$, it is assumed
2.3. SCHEDULABILITY ANALYSIS FOR CAN MESSAGES

that at least one message is transmitted. The message is denoted as $\tau_i$, where $i \in \mathbb{Z}^+$. The properties of $\tau_i$ consist of fixed priority $P_i$, transmission time on the CAN bus $C_i$, queuing jitter $J_i$ and period $T_i$.

The value of $P_i$ is equal to the ID of $\tau_i$, and $\tau_i$ has higher priority than $\tau_j$ if $P_i < P_j$. Since each CAN message has a unique ID in the network, for two messages $\tau_i, \tau_j (i \neq j)$, $P_i \neq P_j$ holds.

$C_i$ is an upper bound on the time it takes to finish transmitting $\tau_i$, which is calculated by Equation 2.1.

Each message is assumed to be queued by a software task, process or interrupt handler executing on the host microprocessor. This task is either invoked by, or polls for, the event and takes a bounded amount of time between 0 and $J_i$ to queue the $\tau_i$ ready for transmission. $J_i$ is referred to as the queuing jitter of the message and is inherited from the overall response time of the task, including any polling delay.

The event that triggers queuing of the message is assumed to occur with a minimum inter-arrival time of $T_i$, referred to as the message period. This model supports events that occur strictly periodically with a period of $T_i$, events that occur sporadically with a minimum separation of $T_i$ and events that occur only once before the system is reset, in which case $T_i$ is infinite.

The WCRT of any message $\tau_i$ is the worst-case delay that $\tau_i$ may experience between arrival and complete transmission, denoted as $R_i$. Each $\tau_i$ is assigned a deadline, $D_i$, which is the longest delay allowed for $\tau_i$. Message $\tau_i$ is schedulable if $R_i \leq D_i$.

2.3.2 THE ANALYSIS METHOD

Schedulability analysis for CAN aims to provide a method of calculating the WCRT of each message. These values can then be compared to the message deadlines to determine if the system is schedulable. For systems complying with the scheduling model given in
section 2.3.1, the WCRT of a message $\tau_i$ is composed of three elements:

1. The queuing jitter $J_i$, corresponding to the longest time between the arrival event and the message being queued, ready to be transmitted on the bus.

2. The queuing delay $\omega_i$, corresponding to the longest time that the message can remain in the CAN station which it belongs to, before transmission.

3. The transmission time $C_i$, corresponding to the longest time that the message can take to be transmitted.

The worst-case response time of $\tau_i$ thus is given by:

$$R_i = J_i + \omega_i + C_i$$

(2.2)

The queuing delay comprises blocking delay, due to lower priority messages which may occupy CAN bus when $\tau_i$ is queued and interference delay due to higher priority messages which may win arbitration and be transmitted in preference to $\tau_i$.

In the worst case, a lower-priority message occupies CAN bus just before $\tau_i$ has been queued, $\tau_i$ thus has to wait whole transmission time of this lower-priority message. Also, in some special cases, the transmission of the previous instance of $\tau_i$ could delay the successive arrived $\tau_i$. For these reasons, the maximum blocking delay $B_i$ is calculated as follows:

$$B_i = \max_{P_k \geq P_i, \tau_k \in \Theta} C_k$$

(2.3)

A further simplification is to assume that the blocking factor always takes its maximum possible value - corresponding to transmission time of the longest possible CAN message (8 data bytes), irrespective of the characteristics and priorities of the messages in the system.
2.3. SCHEDULABILITY ANALYSIS FOR CAN MESSAGES

The interference delay for $\tau_i$ occurs for some instance of $\tau_i$ queued within a priority level-i busy period that starts immediately after the longest lower priority message begins transmission. Busy period is a period of time during which the CAN bus is continually occupied by frame transmission. An extension to this concept, the level $i$ busy period, is defined as a period of time during which the CAN bus is continually occupied by the transmission of messages with priority $P_i$ or higher [11].

The interference delay becomes worst when the level $i$ busy period is longest, which occurs when the busy period begins with a so-called Critical Instant where message $\tau_i$ is queued simultaneously with all higher priority messages and then each of these messages is subsequently queued again after the shortest possible time intervals.

Based on the explanation above, the maximum queuing delay of $\tau_i$ can be calculated by the following equation:

$$\omega_i = B_i + \sum_{\forall k \in hp(i)} \left[ \frac{\omega_i + J_k + t_{bit}}{T_k} \right] C_k$$  \hspace{1cm} (2.4)

where $hp(i)$ is the set of messages with priorities higher than $\tau_i$. Although $\omega_i$ appears on both sides of Equation 2.4, as the right hand side is a monotonic non-decreasing function of $\omega_i$, the equation may be solved using the recurrence relation below.

$$\omega_i^{n+1} = B_i + \sum_{\forall k \in hp(i)} \left[ \frac{\omega_i^n + J_k + t_{bit}}{T_k} \right] C_k$$  \hspace{1cm} (2.5)

A suitable starting value is $\omega_i^0 = B_i$. The relation iterates until $\omega_i^{n+1} = \omega_i^n$, in which case the worst-case response time of the first instance of the message in the busy period is given by: $R_i = J_i + \omega_i^{n+1} + C_i$. 


2.3.3 **Development**

The research of Tindell et al. is based on an assumption: a perfect priority queue can be achieved at each station. Tindell et al. further discussed how a perfect priority queue can be achieved. This research has been completed by A.Meschi et al. [12]. If the priority queue can not be achieved (e.g., due to hardware limitations) [13], one time priority inversion may occur. Natale et al. give a calculation method [14], which is based on the assumption that priority inversion will occur once.

These studies ignored the fact that many commercial MCUs equipped with a CAN controller can support FIFO-queued message transmission (e.g., the M16C/50 series MCU from Renesas [15]). Because FIFO queues may introduce significant priority inversion compared to priority queues, these previous methods cannot work for CAN messages using FIFO queues. To solve this problem, Davis et al proposed a WCRT analysis method for CAN messages using FIFO queues recently [17].

According to these studies, it was known that messages with low priority may be delayed by messages with higher priority, and this becomes worse with the increase of bus load. To mitigate this problem (i.e., reduce the WCRT of low priority message), a method to schedule CAN messages with offsets has been proposed in [7, 8]. Szakaly, Iiyama, as well as Du et al. independently provided WCRT calculation methods based on the priority queue assumption [18, 19, 20]. Their results showed that assigning offset to messages can reduce WCRT by 40%.

### 2.4 Effects of Offset

For a periodic message $\tau_i$ that belongs to station $U_I$, the offset of $\tau_i$ is defined as the interval between the time when $U_I$ is ready to start transmitting messages and the time when the first instance of $\tau_i$ starts to be queued.
To introduce offsets into schedulability analysis, the following notations and assumptions are added into properties of \( \tau_i \).

1. For each \( \tau_i \), \( O_i \) is added to denote offset of \( \tau_i \). Also, constraint \( O_i < T_i \) is assumed to hold.

2. \( A_i^n (n = 0, 1, ...) \) denotes the time when a instance of \( \tau_i \) starts to be queued. Because each ECU transmits messages according to its local time, any \( A_i^n \) is a local time of the station to which \( \tau_i \) belongs.

3. To simplify the analysis, it is assumed that there is no queueing jitter of the messages. Therefore, \( A_i^0 \) is also the time when a instance of \( \tau_i \) is requested to transmit. Based on this assumption, \( A_i^0 \) is called arrival time of a instance of \( \tau_i \) in the remaining context. Hence the first arrival time of \( \tau_i \) equals to its offset, \( A_i^0 = O_i \) is assumed to hold.

The properties of \( \tau_i \) in the offset model are shown in Figure 2.4.

The goal of offset assignment is to decrease the WCRT of messages by avoiding multiple messages being transmitted simultaneously. A comparison of message transmission with and without offsets is shown in Figure 2.5. In the example, \( U_1 \) has two messages \( \tau_1 (P_1 = 1, C_1 = 1, T_1 = 4) \) and \( \tau_2 (P_2 = 2, C_2 = 1, T_2 = 8) \), \( U_2 \) has one message \( \tau_3 (P_3 = 3, C_3 = 1, T_3 = 16) \). In the situation where no offset are assigned (i.e., all messages offsets are 0), \( \tau_1 \) delays \( \tau_2 \), \( \tau_1 \) and \( \tau_2 \) will delay \( \tau_3 \) in the worst-case. On the other
hand, assigning \( O_2 = 2 \) avoids \( \tau_1 \) and \( \tau_2 \) to be transmitted simultaneously, hence WCRT of both \( \tau_2 \) and \( \tau_3 \) is decreased.

2.5 Schedulability Analysis for CAN Messages with Offsets

This section introduces the schedulability analysis for CAN message with offset. The analysis method is summarized based on [18] and improved by considering the flaw pointed out by [9, 10]. First, how the Critical Instant has been re-defined and used for the offset model are shown. Then, the Interference Function (IF) and Maximum Interference Function (MIF) are introduced, which are employed to formulate the WCRT calculation.
Finally, the WCRT calculation formulas are given in the last part of this section.

### 2.5.1 Critical Instant in the Offset Model

In the offset model, the WCRT of a message $\tau_i$ is made up of queuing delay $\omega_i$ and $C_i$, when queuing jitter is assumed zero. $\omega_i$ consists of blocking delay, calculated by Equation 2.3, and interference delay caused by higher-priority messages in self station and other stations.

Note that the classical Critical Instant may be too pessimistic to locate worst case for $\tau_i$ in the offset model, because all messages with higher priority could not be transmitted with $\tau_i$ simultaneously when offsets are assigned. Hence Critical Instant in the offset model has been relaxed to:

“At least one message in each station with a priority higher than $P_i$ is to be released at the critical instant.”

Since it is hard to known which messages will be released at the critical instant, every message with a priority higher than $P_i$ in a station must be treated as a candidate of the critical instant.

Note that the number of worst-case candidates of $\tau_i$ is finite. This is because messages of any station are transmitted repeatedly as the interval of their least common multiple of periods. Thus, only the messages of any station $U_J$, which meet the worst-case candidates conditions and arrive in $[0, LCM_J]$, will be considered in WCRT calculation of $\tau_i$. The $LCM_J$ denotes the least common multiple of periods of all messages in $U_J$. The local time of $U_J$ is used to to describe the simultaneous arrival time of worst-case candidates of $\tau_i$ that belongs to $U_J$. All the simultaneous arrival time that meet the worst-case candidates conditions of $\tau_i$ thus are a set and denoted as $\{WT_i\}$. 
2.5.2 IF\&MIF

To find WCRT in the offset model, *Interference Function (IF)* and *Maximum Interference Function (MIF)* have been employed to formulate the calculation. IF and MIF were first proposed for schedulability analysis in multiframe task model [21]. However, it has been proved that they can also work for schedulability analysis in CAN system [18, 22]. When employed in a CAN system, IF represents the time in an interval in which a set of messages delay lower-priority messages. Thus the delay of $\tau_i$, which is caused by messages of $U_J$ and arrived in $[ST, t]$, can be described by:

$$IF^ST_i[P_i](t)$$

(2.6)

In above expression, $ST$ is the instant at which the IF starts. $[P_i]$ is a condition that only messages arrived in $[ST, t]$ and with priority higher than $P_i$ is included in the IF.

The Maximum Interference Function (MIF) is defined as a function representing the maximum time in an interval, in which a set of messages interferes with the lower-priority message. As the denotation, $M_J[P_i](t)$ describes the maximum time that higher-priority messages of $U_J$ delay $\tau_i$. The calculation of $M_J[P_i](t)$ is given by the following formula:

$$MIF_J[P_i](t) = \max_{\tau_k \in h_{p_J}(i), 0 \leq A^n_k \leq LCM_J} IF^{A^n_k}_J[P_i](t)$$

(2.7)

Here $h_{p_J}(i)$ denotes the set of messages of $U_J$ that have a priority higher than $P_i$.

An example of IF and MIF is given in Figure 2.6 where it is assumed that $\tau_1, \tau_3, \tau_4 \in U_J$, $T_1 = T_3 = T_4 = 8, C_1 = C_3 = C_4 = 1$, and $O_4 = 0, O_1 = 3, O_3 = 4$. Since the arrival of $\tau_1, \tau_3, \tau_4$ in $[0, LCM_J]$ are 0, 3, 4 respectively, $M_J[P_3](t)$ is the max of $IF^0_J[P_3](t)$, $IF^3_J[P_3](t)$ and $IF^4_J[P_3](t)$.

As can be seen from this example, the max operation is used to pick up the uppermost
2.5. SCHEDULABILITY ANALYSIS FOR CAN MESSAGES WITH OFFSETS

Figure 2.6: Examples of $IF$ and $MIF$

line of all $IF$ lines. It is important to note that the maximum interference time derived from $MIF$ is larger than or equal to any that of $IF$. The objective of using $MIF$ instead of $IF$ is to obtain a fast approximate algorithm but not a complex exact algorithm.

$IF$ and $MIF$ support a so-called Saturation Addition that adds multiple $IF$, $MIF$ or transmission time of messages with the maximum slope equal to 1 [22]. The objective of Saturation Addition is to add all the elements that may delay message $\tau_i$ to a new
interference function. Then the first instant at which the slope of the new function becomes 0, is the earliest idle time (EIT) of CAN bus. In other words, it is the first time when \( \tau_i \) can be transmitted to CAN bus. *Saturation Addition* is denoted as ‘\( \oplus \)’ and operation \( EIT(X) \) is used to find the first instant at which the slope of the function \( X \) becomes 0.

An example of *Saturation Addition* and *EIT* is given in Figure 2.7.

### 2.5.3 WCRT Calculation

Based on above explanation, the WCRT of \( \tau_i \) can be calculated as follows.

\[
R_i = \max_{ST \in \{WT_i\}} R_i^{ST}
\]

where

\[
R_i^{ST} = EIT \left( B_{ST} \oplus IF_i^{ST}[P_i](t) \oplus \sum_{UJ \in S, J \neq I} M_J[P_i](t) \right) + C_i + ST - A_i^{ST}
\]  

(2.8)

In the formula, \( A_i^{ST} \) denotes the first arrival time of \( \tau_i \) after \( ST \). \( B_{ST} \) is the blocking delays of the message arriving at \( ST \). \( IF_i^{ST}[P_i](t) \) and \( M_J[P_i](t) \) describe the delays that caused by message of self station \( U_I \) and other station \( U_J \), respectively.
CHAPTER 3

AN OFFSET ASSIGNMENT METHOD FOR IMPROVING SCHEDULABILITY

3.1 OVERVIEW

Although it has been proved that offset can extremely decrease the WCRT of CAN messages, only a few studies focusing on offset assignment have been published. Matsutani et al. first proposed an offset assignment method for CAN messages in [7]. Their method assigns offsets following a policy to keep the simultaneously sending message for each station as small as possible. However, their method does not consider the interference of messages in other stations, therefore it can hardly find satisfactory offset for systems with a high bus load.

Grenier et al. presented their work on offset assignment for CAN messages in [8]. Their approach assigns offsets in such a way that arrivals of any two messages are as separated as possible. Their method improves Matsutani’s approach in that it not only avoids transmission collision in the same station, but also probably mitigates the interference of messages sent by other stations. However, keeping messages as separated as possible is not a direct
approach for decreasing the WCRT of messages.

According to the WCRT calculation method explained in Section 2.5, it is clear that the WCRT of a message $\tau_i$ depends on $MIF_K[P_i](t)$ of all stations $U_K$ in the system. For that reason, decreasing the $MIF_K[P_i](t)$ is the direct way to decrease WCRT for $\tau_i$, and this is the basic approach of the proposed offset assignment method.

Ideally, the best offset assignment — that is, the one that provides the lowest $MIF_K[P_i](t)$ of all $U_K$ for each $\tau_i$ — is expected to be found. In practice, this is usually impossible because decreasing the $MIF_K[P_i](t)$ for $\tau_i$ will generally increase the $MIF_K[P_j](t)$ for $\tau_j$, which means increased WCRT of $\tau_j$. A better approach is to first initialize offsets with such values that the $MIF$ of all messages in each station is as small as possible; and then change the offset to decrease the $MIF_K[P_i](t)$ for message $\tau_i$ that missed its deadlines with the initial offset until the whole system is schedulable.

This chapter is composed of the following sections. Section 3.2 defines terminology and notation that are used in this chapter. Section 3.3 explains a method to initialize the offsets of each message. Section 3.4 describes the algorithm to reform the offsets of messages that were missing their deadlines. Section 3.5 presents the results of several experiments that try to show the effectiveness of the proposed method in comparison with previous work. Finally, this chapter is summarized by Section 3.6.

### 3.2 TERMINOLOGY AND NOTATION

The notation described in Chapter 2 is also used in this chapter. Additionally, new terms — Deadline Rate and Worst-case Response Time Rate — are proposed to evaluate the schedulability of the same system with different offsets. Deadline Rate is defined as the ratio of deadline to period of a message (notice that deadlines are usually defined as a ratio of the period of a message in real automotive systems). Worst-case Response Time
Rate(WCRR) is defined as the ratio of WCRT to period of a message. Denote Deadline Rate and Worst-case Response Time Rate of $\tau_i$ as $DR_i$ and $RR_i$. The formulas to calculate them are given as follows:

$$DR_i = \left(\frac{D_i}{T_i}\right) \times 100\%$$ \hspace{1cm} (3.1)

$$RR_i = \left(\frac{R_i}{T_i}\right) \times 100\%$$ \hspace{1cm} (3.2)

Based on these definitions, the schedulability of a system using an offset assignment method A is higher than the one using an offset assignment method B if the maximum $RR_i$ of method A is lower than that of method B.

### 3.3 Offset Initialization

The goal of offset initialization is to assign offsets to messages so that the MIF of messages in each station is as small as possible. In the proposed method, this is achieved by using the Simulated Annealing (SA) algorithm. The SA algorithm and the offset initialization method based on it are explained in this section.

#### 3.3.1 Simulated Annealing

The SA algorithm is a generic probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum from a given objective function in a large search space [23]. The name and inspiration of SA algorithm come from annealing, a technique used in metallurgy that involves heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more
Algorithm 1 Simulated Annealing Algorithm

(*$S_0$ : Initial Solution*)
(*$OF(S)$ : Objective Function*)
(*$T_0$ : Initial Temperature*)
(*$\alpha$ : Cooling Rate*)

1: $T \leftarrow T_0$, $S_{cur} \leftarrow S_0$, $E_{cur} \leftarrow OF(S)$
2: $S_{best} \leftarrow S_{cur}$, $E_{best} \leftarrow E_{cur}$
3: $k \leftarrow 0$

4: while $k < k_{max}$ do
5:   $S_{new} \leftarrow neighbour(S)$, $E_{new} \leftarrow OF(S_{new})$
6:   $\Delta \leftarrow E_{new} - E_{cur}$
7:   if $\text{random}[0, 1) < \exp(-\Delta/T)$ then
8:       $S_{cur} = S_{new}$, $E_{cur} = E_{new}$
9:   end if
10:  if $E_{new} < E_{best}$ then
11:     $E_{best} \leftarrow E_{new}$
12: end if
13:  $k \leftarrow k + 1$, $T \leftarrow \alpha T$
14: end while
15: return $S_{best}$

chances of finding configurations with lower internal energy than the initial one. The $SA$ aims to find an acceptably good solution in a fixed amount of time, rather than trying to find the best possible solution. Hence it is more efficient than exhaustive enumeration.

The pseudocode of the $SA$ algorithm is given in Algorithm 1. The algorithm starts with an initial solution $S_0$ and performs a maximum of $k_{max}$ steps. At each step, it calls $neighbour(S)$ to generate a randomly chosen new solution $S_{new}$ and probabilistically decides whether or not to replace $S_{cur}$ with $S_{new}$. The probability is usually decided using the $Metropolis$ method [24] which accepts $S_{new}$ depending on the difference between $E_{new}$ and $E_{cur}$, as well as on a global parameter $T$. The $T$ denotes the temperature, and it is gradually decreased with $\alpha(0 < \alpha < 1)$ during the algorithm. If the $objective function$ value of $S_{new}$ is smaller than that of $S_{cur}(E_{new} < E_{cur})$, $S_{new}$ is selected as a temporary solution. Otherwise, whether or not selecting $S_{new}$ will be determined randomly. In general, the probability of selecting $S_{new}$ will be decreased as the temperature $T$ is reduced.
This feature of SA is effective to move away from a local optimal solution and improve the probability of finding the global optimal solution. Finally, $S_{best}$ corresponding to the $E_{best}$ is generated as the best solution.

3.3.2 THE PROPOSED OFFSET INITIALIZATION ALGORITHM

Based on the purpose of the proposed offset assignment method, which keeps $MIF$ of messages in each station as small as possible, the Integration of Maximum Interference Function (IMIF) of each station is selected as the objective function of SA algorithm. Denote $MIF$ of all messages of $U_I$ as $MIF_I$, $IMIF$ of $U_I$ as $IMIF_I$, $IMIF_I$ is calculated by the following equation:

$$
IMIF_I = \int_0^{LCM_I} MIF_I(t) dt \quad (3.3)
$$

The algorithm that initializes offsets for messages in each station $U_I$ — based on the SA algorithm — is detailed below:

1. Initialize the parameters $T$ (temperature), $k$ (current searching steps) and $k_{max}$ (maximum searching steps).

2. Assign random offsets to messages in station $U_I$ and save them in $S_{cur}$ and $S_{best}$. The $IMIF_I$ of these offsets are calculated and stored in $E_{cur}$ and $E_{best}$.

3. A neighbor function is used to randomly select a message $\tau_j (\tau_j \in U_I)$, and randomly change $O_j$ to $O_j + 1$ or $O_j - 1$ and save them in $S_{new}$. The $IMIF_I$ is recalculated and stored in $E_{new}$, $\Delta$ is calculated as $E_{new} - E_{cur}$.

4. If $\text{random}[0, 1] < \exp(-\Delta/T)$, then update $S_{cur}$ and $E_{cur}$ with $S_{new}$ and $E_{new}$, respectively.
5. Update $S_{\text{best}}$ with $S_{\text{new}}$ if $E_{\text{new}} < E_{\text{best}}$.

6. Cool down the temperature $T$ and increase $k$. Repeat the process from step (3) if $k < k_{\text{max}}$. Otherwise, finish the process and the offsets corresponding to $S_{\text{best}}$ are selected as the offsets of messages in $U_1$.

### 3.4 Offset Reformation

First of all, an example is given to explain the need for offset reformation. Focusing on the message set given in Table 3.1, in which $U_1$ has three messages $\tau_1, \tau_2, \tau_4$, $U_2$ has one message $\tau_3$. The $DR_i$ of all messages is assumed to be 60%. Assigning offsets to messages in $U_1$ as $O_1^i$ based on the algorithm given in 3.3, the lower bound $IMIF_1$ is obtained. However, $\tau_3$ misses its deadline since $RR_1^3$ is 75%. On the other hand, $\tau_3$ can meet its deadline in the offset assignment $O_2^3$, which is set by keeping $IMIF_1[P_3]$ — the $IMIF$ that only includes messages with higher priority than $\tau_3$ — as small as possible. A comparison of the $IMIF$ corresponding to $O_1^3$ and $O_2^3$ is shown in Figure 3.1. It is observed that reforming the offset for a message $\tau_i$ — by keeping $IMIF_1[P_i]$ as small as possible — can further decrease $WCRT$ of $\tau_i$.

In order to achieve the offset reformation, the objective function of the $SA$ algorithm

<table>
<thead>
<tr>
<th>$U_1$</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_i$</td>
<td>$P_i$</td>
<td>$E_i$</td>
<td>$C_i$</td>
<td>$O_1^i$</td>
<td>$O_2^i$</td>
</tr>
<tr>
<td>$\tau_1$</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$\tau_4$</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$U_2$</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_i$</td>
<td>$P_i$</td>
<td>$E_i$</td>
<td>$C_i$</td>
<td>$R_1^i(RR_1^i)$</td>
<td>$R_2^i(RR_2^i)$</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>6(75%)</td>
<td>4(50%)</td>
</tr>
</tbody>
</table>
3.4. OFFSET REFORMATION

is changed from Equation 3.3 to Equation 3.4 as follows:

\[ IMIF'_I = IMIF_I + \sum_i W[P_i] \ast IMIF_I[P_i] \]  

(3.4)

In the equation, \( W[P_i] \) denotes the Weighting Factor (WF) of \( \tau_i \), which controls weighting of \( \tau_i \) in the offset reformation. Using the \( IMIF' \), the complete offset assignment algorithm is given as follows:

1. Set the \( WF \) of all messages to 0. Set the maximum execution steps of the program to \( n_{max} \).
2. Set $IMIF'$ as the objective function of the SA algorithm, and call the proposed SA algorithm described in Section 3.3.2, and assign the obtained offsets to messages in each station.

3. Calculate the WCRT for all messages and find the deadline-missed message ($\tau_k$) with the largest worst-case response time rate.

4. Add 1 to $W[P_k]$. Then go back to step(2) and repeat the process until all messages meet their deadlines or the execution steps exceeds $k_{max}$.

An example is given to explain the algorithm in details. In the example, $U_1$ has messages $\tau_1, \tau_2, \tau_5, \tau_6$; $U_2$ has messages $\tau_3, \tau_4, \tau_7, \tau_8$; the priority for these messages are assumed to follow $P_1 < P_2 < \ldots < P_8$. First, it is assumed that $\tau_4, \tau_7$ miss their deadlines with the initialized offsets ($WF$ of all messages are 0), and $RR_4 > RR_7$. $W[P_4]$ is set to 1 before running the offset reformation. The objective function thus becomes to:

$$IMIF'_I = IMIF_I + 1 \times IMIF_I[P_4].$$

Next, assume that $\tau_4, \tau_7$ still miss their deadlines and $RR_4 > RR_7$ after the first-time offset reformation. $W[P_4]$ is then set to 2 before the second-time offset reformation and the objective function becomes:

$$IMIF'_I = IMIF_I + 2 \times IMIF_I[P_4].$$

$\tau_4$ is assumed to meet its deadline after the second-time offset reformation. Then, focus on $\tau_7$, $W[P_7]$ is set to 1 and the objective function for the third-time offset reformation becomes:

$$IMIF'_I = IMIF_I + 2 \times IMIF_I[P_4] + 1 \times IMIF_I[P_7].$$

3.5 Evaluation

The proposed method is evaluated in this section by using real-world message sets provided by automakers and message sets generated by NETCARBENCH [26]. In the experiments, first, offsets are initialized by using the algorithm given in 3.3($WF$ of all messages are
Next, a deadline rate parameter is set, which is used as the deadline for conducting the offset reformation. The deadline rate parameter is first chosen as a value that is smaller than the maximum $RR_i$ of the message set with the initialized offset. Then, the deadline rate parameter is decreased and the offset reformation is repeated until the program cannot find offsets that meet the time constraints of all messages.

Furthermore, to show the effectiveness of the proposed method, comparisons between the proposed method and the previous method [8] are conducted. The method [8] is briefly summarized as follows:

1. Initialize an empty offset assignment record for all messages.

2. In each $U_l$, find the $\tau_i$ with the shortest period message in station $U_l$ that has not been assigned an offset.

3. Based on the offset assignment record, find the longest interval $(t_0, t_1)$ of $[0, T_i)$ in which no message arrives.

4. Assign $(t_0 + t_1)/2$ to $O_i$ and update the offset assignment record.

5. Repeat steps (2) to (4) until all messages have their offsets assigned.

All the experiments are performed on a computer with an Intel Core i5 2.67GHz processor. The proposed method finishes the calculation of each experiment in 30 minutes; the previous method finishes the calculation of each experiment in 5 minutes.

### 3.5.1 Experiments on Message Sets Provided by Automakers

First, offsets are assigned to a message set provided by an automotive maker. The system has 15 stations, 65 messages and 35% bus load. The comparison with the previous method is shown in Table 3.2. It can be seen that offsets initialized using the proposed method
decrease the average and maximum $RR_i$ by 4.91% and 11.37% respectively, compared to the previous method.

Then, the deadline rate parameter is set as 32.3% and run the offset reformation. It is observed that the average and maximum $RR_i$ were decreased by 4.54% and 8.24% respectively. In particular, all messages managed to meet their deadlines after the proposed offset assignment.

Table 3.2: Experiment 3.1: The message set is provided by an automaker for a system with 15 stations, 65 messages and 35% bus load

<table>
<thead>
<tr>
<th>Offset Assignment</th>
<th>Average $RR_i$ (Decrease Rate)</th>
<th>$\max RR_i$ (Decrease Rate)</th>
<th>Number of Deadline Miss Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Method</td>
<td>8.33% ( - )</td>
<td>35.20% ( - )</td>
<td>1</td>
</tr>
<tr>
<td>Proposed Method</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initialization</td>
<td>8.03% (3.66%)</td>
<td>32.50% (7.67%)</td>
<td>1</td>
</tr>
<tr>
<td>Reformation (Once)</td>
<td>7.95% (4.54%)</td>
<td>32.30% (8.24%)</td>
<td>0</td>
</tr>
</tbody>
</table>

In the second experiment, offsets are assigned to a message set from another automaker whose system consists of 14 stations, 68 messages with 65% bus load. As shown in Table 3.2, offsets initialized by the proposed method decrease the average and maximum $RR_i$ by 4.91% and 11.37%, compared to the previous method. Then, the deadline rate parameter is set as 32.5% and reform the offset. A double reform decreases the average and maximum $RR_i$ by 7.66% and 14.86% respectively. It is observed that all messages were able to meet their deadlines after the double reform.

### 3.5.2 Experiments on Message Sets Generated by NETCAR-Bench

In the experiments described above, message sets have a large number of stations; and the bus load of messages in each station is relatively low. This limits the effectiveness of offset
3.5. EVALUATION

Table 3.3: Experiment 3.2: The message set is provided by another automaker for a system with 14 stations, 68 messages and 65% bus load

<table>
<thead>
<tr>
<th>Offset Assignment</th>
<th>Average $RR_i$ (Decrease Rate)</th>
<th>max $RR_i$ (Decrease Rate)</th>
<th>Number of Deadline miss Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Method</td>
<td>18.95% ( - )</td>
<td>38.13% ( - )</td>
<td>1</td>
</tr>
<tr>
<td>Proposed Method</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initialization</td>
<td>18.02% (4.91%)</td>
<td>33.79% (11.37%)</td>
<td>1</td>
</tr>
<tr>
<td>Reformation (Once)</td>
<td>17.49% (7.70%)</td>
<td>32.71% (14.21%)</td>
<td>1</td>
</tr>
<tr>
<td>Reformation (Twice)</td>
<td>17.50% (7.66%)</td>
<td>32.46% (14.86%)</td>
<td>0</td>
</tr>
</tbody>
</table>

assignment. For that reason, additional experiments are conducted on two message sets generated by NETCARBENCH with high bus load in each station.

In the third experiment, a message set is generated for a system with 5 stations, 75 messages and 55% bus load. As shown in Table 3.4, compared to the previous method, offsets initialized using the proposed method decrease the average $RR_i$ by 16.13%, the maximum $RR_i$ by 3.29% and the number of messages missing their deadlines are reduced from 5 to 2 with the deadline rate parameter set to 15.00%. After running the offset reformation algorithm with this deadline rate, the average and maximum $RR_i$ were decreased by 16.49% and 9.88%, compared to the previous method; and all messages were able to meet their deadlines after the offset reformation.

Table 3.4: Experiment 3.3: The message set is generated by NETCARBENCH for a system with 5 stations, 75 messages and 55% bus load.

<table>
<thead>
<tr>
<th>Offset Assignment</th>
<th>Average $RR_i$ (Decrease Rate)</th>
<th>max $RR_i$ (Decrease Rate)</th>
<th>Number of Deadline miss Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Method</td>
<td>6.59% ( - )</td>
<td>16.40% ( - )</td>
<td>5</td>
</tr>
<tr>
<td>Proposed Method</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initialization</td>
<td>5.53% (16.13%)</td>
<td>15.86% (3.29%)</td>
<td>2</td>
</tr>
<tr>
<td>Reformation (Once)</td>
<td>5.50% (16.49%)</td>
<td>14.78% (9.88%)</td>
<td>0</td>
</tr>
</tbody>
</table>
In the last experiment, a message set is generated for a system with 5 stations, 74 messages and 65% bus load. The comparison between the proposed method and previous method is shown in Table 3.5. It is observed that offsets initialized using the proposed method decreased average and maximum $RR_i$ by 25.69% and 28.93%, compared to the previous method. A double reform further decreased the average and maximum $RR_i$ by 32.80% and 35.94%. The number of messages missing their deadlines is first decreased from 11 to 5 by the offset initialization method; and then it is further decreased to 0 by the offset reformation.

Table 3.5: Experiment 3.4: The message set is generated by NETCARBENCH for a system with 5 stations, 74 messages and 65% bus load

<table>
<thead>
<tr>
<th>Offset Assignment</th>
<th>Average $RR_i$ (Decrease Rate)</th>
<th>max $RR_i$ (Decrease Rate)</th>
<th>Number of Deadline miss Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Method</td>
<td>10.83% (-)</td>
<td>29.38% (-)</td>
<td>11</td>
</tr>
<tr>
<td>Proposed Method</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initialization</td>
<td>8.04% (25.69%)</td>
<td>20.88% (28.93%)</td>
<td>5</td>
</tr>
<tr>
<td>Reformation (Once)</td>
<td>7.69% (29.01%)</td>
<td>19.80% (32.61%)</td>
<td>2</td>
</tr>
<tr>
<td>Reformation (Twice)</td>
<td>7.27% (32.80%)</td>
<td>18.82% (35.94%)</td>
<td>0</td>
</tr>
</tbody>
</table>

### 3.6 SUMMARY

This chapter explained the proposed offset assignment method for CAN messages. According to the experiments with 4 message sets, it is observed that the proposed method was able to decrease the average $RR_i$ by 4% to 33%, and the maximum $RR_i$ by 3% to 35%, compared to a previous method. Specially, in the situation that the message set has high bus load of messages in each station, the proposed method shows more advantages. Therefore, the proposed method can effectively improve the schedulability of CAN messages. Although calculation time of the proposed method is more than that of the previous
method (in the experiments which are shown in Section 3.5, the proposed method finished the calculation in 30 minutes, the previous method finished the calculation in 5 minutes), a calculation time of 30 minutes is acceptable for a real world offset assignment application.
CHAPTER 4

WCRT ANALYSIS FOR FIFO-BASE
OFFSET ASSIGNED CAN MESSAGES

4.1 OVERVIEW

As mentioned in the Section 2.3.3, most of the previous studies focus on the WCRT analysis based on the system using a priority queue. However, many commercial micro controller unit (MCU) equipped with CAN controller also provide the FIFO queue to transmit message (e.g., the M16C/50 series MCU from Renesas [15]). Because the FIFO queue has faster queue management, using FIFO queue in CAN system can seem an attractive solution to improve the performance of the system [16]. However, to the best of my knowledge, only one published research discussed the WCRT analysis method of FIFO queued CAN message[17] and this study does not consider offset. For this reason, a WCRT analysis method is proposed for the FIFO-based offset assigned CAN messages.

In this chapter, first, a critical instant theorem to locate the worst case of a given message is presented. Then, based on the theorem two algorithms are proposed to calculate the WCRT. Specifically, an exact algorithm is proposed for accurate calculation, and an ap-
propane algorithm is proposed for rapid estimation. The exact algorithm demands high computational complexity, which is suitable for evaluation of the approximate algorithm or for calculations in small systems. In contrast, the approximate could estimate WCRT with limited errors and low computational complexity. And it is expected to be useful in large systems. It is proved that the exact algorithm is accurate and the approximate algorithm is sufficiently safe for WCRT analysis. Furthermore, experiments are conducted to validate the above two algorithms by using generated message sets and a real message set from an automaker.

This chapter is organized as follows. Section 4.2 describes the system models, terminology and notation that are used for the proposed schedulability analysis method. Section 4.3 explains the new critical instant theorem. Section 4.4 presents the algorithms for WCRT calculation. Section 4.5 shows the experiments and results. The extension with consideration of jitter is explained in Section 4.6, followed by a summary in Section 4.7.

4.2 SYSTEM MODEL, TERMINOLOGY AND NOTATION

4.2.1 MODELING

The system for analysis is denoted by \( \Theta \) that consists of a CAN bus and multiple CAN stations. Each station of \( \Theta \) is denoted as \( U_I \), where \( I \in \mathbb{Z}^+ \). Because message transmission is synchronized in same station but asynchronous in different stations, It is assumed that the phase \( \phi_I (\phi_I \geq 0) \) occurs between the start time of the network and that of the station \( U_I \). Since each station can be started at any instant after network is initialized, \( U_I \) can have any \( \phi_I \). For convenience of analysis, it is assumed that the network has a global system clock and time \( 0 \) is defined as the instant when the CAN bus is ready to transmit messages. For each \( U_I \), it is assumed that at least one message is transmitted. The message is denoted
as $\tau_i$, where $i \in \mathbb{Z}^+$. The properties of $\tau_i$ consist of fixed priority $P_i$, transmission time on the CAN bus $C_i$, and period $T_i$.

All messages in the system are defined as periodic messages and reoccur infinitely. A frame is defined as each occurrence of a message. The frames of $\tau_i$ are $\tau_{i,0}, \tau_{i,1}, \ldots$. Define arrival and start are the instants of network time when a frame is requested to transmit, and starts to be transmitted from the queue to the CAN bus, respectively. Arrival and start of $\tau_{i,m}$ are denoted as $a_{i,m}$ and $s_{i,m}$. $O_i$ thus is the interval between $\phi_i$ and $a_{i,0}$ (i.e., the arrival of the first frame of $\tau_i$) and $T_i$ is the interval between $a_{i,m}$ and $a_{i,m+1}$. Figure 4.1 is an example showing each property of a message $\tau_i$.

The level $i$ busy period is denoted by $\beta_{i}$. A frame $\tau_{i,m}$, which arrives during $\beta_{i-1}$, will be able to gain access to the CAN bus after the end of the $\beta_{i-1}$. The WCRT of any frame $\tau_{i,m}$ is the worst-case delay that $\tau_{i,m}$ may experience between arrival and complete transmission. Denote the WCRT of $\tau_{i,m}$ as $R_{i,m}$. The maximum $R_{i,m}$ is thus the WCRT of $\tau_i$, denoted as $R_i$. Each $\tau_i$ is assigned a deadline, $D_i$, which is the longest delay allowed for each frame $\tau_{i,m}$. It is assumed that $D_i \leq T_i$ holds for each message $\tau_i$.  

![Figure 4.1: Properties of message $\tau_i$ for FIFO-queue based analysis](image)
4.2.2 Assumptions

To simplify the analysis, it is assumed that there is no jitter on the arrivals of the messages (the extension with consideration of jitter is explained in Section 4.6). For the same purpose, all stations of the system are assumed to use FIFO to queue frames. Note that system consisting of stations with FIFO or priority queue can be analyzed by extending the proposed method.

Also, it is assumed that in the FIFO queue the earlier a frame arrives, the earlier the frame can attempt to be transmitted to the CAN bus. However, if messages of the same station arrive at the same instant, it is assumed that the higher-priority message is queued earlier.

4.3 The Worst Case of Messages in FIFO Queues

According to the previous research, calculating the WCRT of a message can be achieved by the following two steps:

1. Locate the worst case of the message.

2. Calculate the response time of the message in its worst case.

The critical instant was employed to locate the worst case of messages in priority-queue used system. It is defined as the instant at which a message arrival will have the largest response time. However, the FIFO-based and offset assigned system is too complex to be analyzed by the former critical instant definition. To solve this problem, a redefined critical instant is given and a critical instant theorem to locate the worst case of the FIFO queued message is presented.

In the remainder of this section, Section 4.3.1 presents several definitions then explains redefined critical instant. Section 4.3.2 describes the critical instant theorem of the FIFO-
4.3. THE WORST CASE OF MESSAGES IN FIFO QUEUES

4.3.1 DEFINITIONS OF CRITICAL INSTANT AND CRITICAL INSTANT CANDIDATE

Definition 4.1 (Effect). Effect describes the delay of a frame, which is related with the earlier queued frame in the same station.

Assume two frames \( \tau_{i,m}, \tau_{j,n} \in U_I \), and \( \tau_{i,m} \) is queued earlier than \( \tau_{j,n} \). Then, \( \tau_{i,m} \) affects \( \tau_{j,n} \) if one of the following conditions hold:

1. The instant, at which \( \tau_{i,m} \) is completely transmitted, is later than \( a_{j,n} \).

2. A busy period \( \beta_{j-1} \) continually occupies the CAN bus after \( \tau_{i,m} \) is completely transmitted, and the finish of \( \beta_{j-1} \) is later than \( a_{j,n} \).

The above two conditions are illustrated in Figure 4.2.

Definition 4.2 (Successive Frame Sequence). Assume two frames \( \tau_{i,m}, \tau_{j,n} \in U_I \), and \( \tau_{i,m} \) is queued earlier than \( \tau_{j,n} \). If no other frame in \( U_I \) is queued between \( \tau_{i,m} \) and \( \tau_{j,n} \), then \( \tau_{j,n} \) is the successive frame of \( \tau_{i,m} \).

Because queuing order is important for analysis, \( \Gamma_0, \Gamma_1, \ldots, \Gamma_n \) is used to denote a successive frame sequence that are sorted by the queued order.
**Definition 4.3** (Successive Affect Frame Sequence). In successive frames \( \Gamma_i, \Gamma_{i+1}, \ldots, \Gamma_{i+m} \), if \( \Gamma_i \) affects \( \Gamma_{i+1}, \Gamma_{i+1} \) affects \( \Gamma_{i+2}, \ldots, \Gamma_{i+m-1} \) affects \( \Gamma_{i+m} \), then \( \Gamma_i, \Gamma_{i+1}, \ldots, \Gamma_{i+m} \) is the \( \Gamma_i \) successive affect frame sequence.

The \( \Gamma_i \) successive affect frame sequence is denoted as \( \Gamma_{Ai-seq} \). Any frame \( \Gamma_{i+m} \) in \( \Gamma_{Ai-seq} \) is denoted by \( \Gamma_{A(i,m)} \). Also, \( P_{A(i,m)}, C_{A(i,m)}, a_{A(i,m)}, s_{A(i,m)} \) and \( R_{A(i,m)} \) denote the priority, transmission time on the CAN bus, arrival, start, and worst-case response time of \( \Gamma_{A(i,m)} \), respectively. In addition, the level \( P_{A(i,m)} \) busy period is denoted as \( \beta_{A(i,m)} \), thus frame \( \Gamma_{A(i,m)} \) that arrives in \( \beta_{A(i,m)}-1 \) can only access the CAN bus after the finish of \( \beta_{A(i,m)}-1 \).

**Definition 4.4** (Critical Instant). The critical instant of \( \Gamma_{A(i,m)} \) is defined as the instant at which the arrival of \( \Gamma_{A(i,0)} \) will lead to the largest response time of \( \Gamma_{A(i,m)} \).

Note that this definition differs from the previous one in two ways. First, the critical instant is for a frame but not a message. Second, the frame is involved in a successive affect frame sequence. However, in the offset assigned system, it is difficult to find the critical instant of \( \Gamma_{A(i,m)} \) directly. Therefore, the following definition is given to locate the critical instant of \( \Gamma_{A(i,m)} \).

**Definition 4.5** (Critical Instant Candidates (CICs) of \( \Gamma_{A(i,m)} \)). Assume \( \Gamma_{A(i,l_0)} \) is the lowest-priority frame queued between \( \Gamma_{A(i,0)} \) and \( \Gamma_{A(i,m)} \). The CICs of \( \Gamma_{A(i,m)} \) are defined as the instants that match the following conditions:

1. \( \Gamma_{A(i,0)} \), which is the first frame in \( \Gamma_{Ai-seq} \), arrives simultaneously with any one of the frames belonging to the other stations with a priority higher than \( \Gamma_{A(i,l_0)} \).

2. A frame with a priority lower than \( \Gamma_{A(i,0)} \), belonging to the other stations and having the largest transmission time, occupies the CAN bus just before the arrival of \( \Gamma_{A(i,0)} \).
4.3. THE WORST CASE OF MESSAGES IN FIFO QUEUES

Table 4.1: A message set example

<table>
<thead>
<tr>
<th></th>
<th>(U_i)</th>
<th>(\tau_i)</th>
<th>(P_i)</th>
<th>(T_i)</th>
<th>(C_i)</th>
<th>(O_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(U_1)</td>
<td>(\tau_8)</td>
<td>(P_8)</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(U_2)</td>
<td>(\tau_5)</td>
<td>(P_5)</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\tau_6)</td>
<td>(P_6)</td>
<td>8</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\tau_2)</td>
<td>(P_2)</td>
<td>8</td>
<td>1</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>(U_3)</td>
<td>(\tau_4)</td>
<td>(P_4)</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\tau_1)</td>
<td>(P_1)</td>
<td>8</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>(U_4)</td>
<td>(\tau_7)</td>
<td>(P_7)</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\tau_3)</td>
<td>(P_3)</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.3: An example of a critical instant candidate

An example is given to explain the CICs. Assume a system consists of 4 stations and 8 messages, whose information is shown in Table 4.1. Focus on the WCRT analysis of the frame \(\tau_{3,0}\) in \(U_4\). Since \(\tau_{7,0}\) is first queued before \(\tau_{3,0}\) in the \(U_4\), \(\tau_{7,0}\) and \(\tau_{3,0}\) are the successive affect frame sequence, which can be represented by \(\Gamma_{A(0,0)}\) and \(\Gamma_{A(0,1)}\), respectively. In particular, the \(\tau_{7,0}\) can also be represented by \(\Gamma_{A(1,0)}\) because it is the lowest-priority frame. From the definition above it is clear that \(\tau_{3,0}\) has 6 CICs which meet the following conditions on arrival time: \((a_{5,0} = a_{4,0} = a_{7,0}), (a_{6,0} = a_{4,0} = a_{7,0}), (a_{2,0} = a_{4,0} = a_{7,0}),\)
(a_{5,0} = a_{1,0} = a_{7,0}), (a_{6,0} = a_{1,0} = a_{7,0}), (a_{2,0} = a_{1,0} = a_{7,0})$, respectively. As a concrete example, one of the $CIC$s is depicted in Figure 4.3. The example corresponds to the above condition $(a_{5,0} = a_{4,0} = a_{7,0})$, which means that $\tau_{5,0}, \tau_{4,0}, \tau_{7,0}$ arrive simultaneously at network time 0. Note that $\tau_{8,0}$ is the frame that meets the second condition of $CIC$s definition, thus it is assumed to occupy the CAN bus just before the network time 0.

4.3.2 Critical Instant Theorem

Because it is important to guarantee that the maximum response time of $\Gamma_{A(i,m)}$ always exists in its $CIC$s, the critical instant theorem is given and proved as follows.

**Theorem 4.1.** The critical instant of $\Gamma_{A(i,m)}$ occurs at one of the $CIC$s of $\Gamma_{A(i,m)}$.

**Proof.** The proof is achieved by following steps:

1. Assume $S_1$ is any situation. $S_2$ is a variation of $S_1$, in which frames of $U_I$ match the $CIC$s condition. $S_3$ is a variation of $S_2$, in which frames of $U_J (J \neq I)$ match the $CIC$s condition either. $S_4$ is a variation of $S_3$, in which frames of all the stations match the $CIC$s condition.

2. It is proved that $S_2$ can always be found, in which response time of $\Gamma_{A(i,m)}$ is large than or equal to it in $S_1$. Then, $S_3$ can always be found, in which response time of $\Gamma_{A(i,m)}$ is large than or equal to it in $S_2$. Finally, $S_4$ can always be found too, in which response time of $\Gamma_{A(i,m)}$ is large than or equal to it in $S_3$.

3. Because all $S_4$ are included in the $CIC$s of $\Gamma_{A(i,m)}$ and response time of $\Gamma_{A(i,m)}$ is large than or equal to it in $S_1$, the proof is achieved.

In detail, It is assumed that frames of $\Gamma_{A_{i-seq}}$ belong to $U_I$, and there is any situation $S_1$, in which $\Gamma_{A(i,0)}$ arrives at $t_1$, and $\Gamma_{A(i,m)}$ finishes at $t_e$ as shown in Figure 4.4. Also, assume that $t_0$ is the last instant before $a_{A(i,0)}$ at which no frame with a priority higher than $\Gamma_{A(i,l_0)}$
4.3. THE WORST CASE OF MESSAGES IN FIFO QUEUES

is transmitted on the CAN bus. Meanwhile, \( \Gamma_{A(i,0)} \) is the lowest-priority frame between \( \Gamma_{A(i,0)} \) and \( \Gamma_{A(i,m)} \). Because no frame is transmitted at network time 0, network time 0 meets the assumption of \( t_0 \). In other words, \( t_0 \) always exists.

Because the interval, from \( t_0 \) to the instant when \( \Gamma_{A(i,0)} \) starts to be transmitted, is occupied by frames with priorities higher than \( \Gamma_{A(i,0)} \), this interval is busy period \( \beta_{A(i,0)} \). Again, search the lowest-priority frame \( \Gamma_{A(i,1)} \), which is queued between \( \Gamma_{A(i,0)} \) and \( \Gamma_{A(i,m)} \). Then, the interval, from the instant when \( \Gamma_{A(i,0)} \) is completely transmitted to the instant when \( \Gamma_{A(i,1)} \) starts to be transmitted, is the busy period \( \beta_{A(i,1)} \). Continue this searching until \( \Gamma_{A(i,m)} \) becomes the lowest-priority frame after \( \Gamma_{A(i,m-1)} \). Let \( \Gamma_{A(i,n)} \) be equal to \( \Gamma_{A(i,m)} \), the interval \([t_0, t_e]\) can be described by \( \beta_{A(i,0)} \), \( \beta_{A(i,1)} \), ..., \( \beta_{A(i,n)} \), as shown in Figure 4.4.

Assume that \( U_I \) starts earlier in situation \( S_2 \), so that \( \Gamma_{A(i,0)} \) arrives at \( t_0 \) as shown in Figure 4.5. The arrivals of frames in other stations do not change. Compared with \( S_1 \), the length of each busy period in \([t_0, t_e]\) does not change in \( S_2 \). \( \Gamma_{A(i,n)} \) still finishes transmission at \( t_e \). However, since \( t_0 \leq t_1 \), the arrivals of frames \( \Gamma_{A(i,0)}, \ldots, \Gamma_{A(i,m)} \) in \( S_2 \) are earlier than or equal to that in \( S_1 \). Thus, the response time of \( \Gamma_{A(i,n)} \) in \( S_2 \) is longer than or equal to that in \( S_1 \).

Assume that \( t_2 \) is an instant at which a frame of \( U_J (J \neq I) \) first arrives after \( t_0 \) in \( S_2 \) as shown in the \( S_2 \) of Figure 4.6. Also assume that the frame of \( U_J \) first arrives at \( t_0 \) in situation \( S_3 \), as shown in the \( S_3 \) of Figure 4.6. The following changes occur from \( S_2 \) to \( S_3 \):
Figure 4.5: From situation $S_1$ to $S_2$

1. Since $P_{A(i,l_0)} \geq P_{A(i,l_1)} \cdots \geq P_{A(i,l_n)}$, for each $\beta_{A(i,l_0)} - 1$, the frames included in $\beta_{A(i,l_0)} - 1$ in $S_2$ still exist in $\beta_{A(i,l_0)} - 1$ or $\beta_{A(i,l_0)} - 1$ in $S_3$. Thus, no matter what the situation is, the sum of the lengths of all $\beta_{A(i,l_0)} - 1$ does not change. $\Gamma_{A(i,m)}$ (i.e., $\Gamma_{A(i,l_n)}$) will still finish transmission at $t_e$.

2. The frames of $U_f$, which arrive after $\Gamma_{A(i,l_0)}$, with a priority lower than $\Gamma_{A(i,l_0)}$ but higher than $\Gamma_{A(i,l_0)}$, do not exist in any busy period in $S_2$, but may exist in $\beta_{A(i,l_0)} - 1$ in $S_3$. For example, as shown in Figure 4.6, the frame of $U_f$, which arrive after $\Gamma_{A(i,l_1)}$ with a priority lower than $\Gamma_{A(i,l_2)}$ but higher than $\Gamma_{A(i,l_1)}$, does not exist in any busy period in $S_2$, but exist in the $\beta_{A(i,l_1)} - 1$ in $S_3$. For this reason, the sum of the lengths of all $\beta_{A(i,l_0)} - 1$ in $S_3$ may be larger than that in $S_2$. $\Gamma_{A(i,m)}$ will be transmitted completely at $t_e'$ in $S_3$, which is later than $t_e$ in $S_2$.

Finally, in situation $S_4$, the first frame in $[t_0, t_e']$ of every other $U_K$ arrives at $t_0$. And, a frame of other stations with a priority lower than $\Gamma_{A(i,0)}$ and the largest transmission time occupies the CAN bus just before $t_0$. Then, $S_4$ is a CIC of $\Gamma_{A(i,m)}$. The response time of $\Gamma_{A(i,m)}$ in $S_4$ is larger than or equal to that in $S_1$.

According to the above results, it is known that for any situation $S_1$, a relative situation $S_4$ can always be found, in which the response time of $\Gamma_{A(i,m)}$ is larger than or equal to that
4.4. THE PROPOSED ALGORITHMS FOR WCRT CALCULATION

According to the above theorem, the WCRT of messages in the station $U_I$ can be calculated by the following steps:

1. Define $LCM_I$ as the least common multiple of periods of all messages in $U_I$. For each $\Gamma_i (\Gamma_i \in U_I, \Gamma_i$ arrives between $\phi_I$ and $\phi_I + LCM_I)$, focus on its successive affect frame sequence $\Gamma_{Ai-seq}$.

2. For each $\Gamma_{A(i,m)} (m = 0, 1, \ldots)$ of each $\Gamma_{Ai-seq}$, locate all $CIC$s of $\Gamma_{A(i,m)}$.

3. Calculates $R_{A(i,m)}$ from these CIC$s.
4. Compare $R_{A(i,m)}$ with WCRT of the message which $\Gamma_{A(i,m)}$ belongs to. Update WCRT of this message if $R_{A(i,m)}$ is larger.

In this method, the step 3 that calculates the WCRT for a given frame is the most important. For this calculation, an exact algorithm and an approximate algorithm are proposed as follows.

4.4.1 THE EXACT ALGORITHM

To obtain the exact WCRT of a given frame, a method is to calculate the $R_{A(i,m)}$ by checking all the CICs of $\Gamma_{A(i,m)}$ as the given theorem. The key to achieve this goal is to determine the latest finish of $\beta_{A(i,m)} - 1$, at which the $\Gamma_{A(i,m)}$ will be able to transmit on the CAN bus. In a simple situation, $\Gamma_{A(i,m)}$ is the lowest-priority frame between $\Gamma_{A(i,0)}$ and $\Gamma_{A(i,m)}$, so that only one busy period $\beta_{A(i,m)} - 1$ needs to be considered. However, as shown in Figure 4.4, in general case multiple busy periods should be considered. In addition, the start of each $\beta_{A(i,l_x) - 1}$ is related to the finish of $\beta_{A(i,l_{x-1}) - 1}$. Thus, to find the latest finish of $\beta_{A(i,m) - 1}$, it is necessary to calculate the maximum sum of the lengths of all busy periods between $\Gamma_{A(i,0)}$ and $\Gamma_{A(i,m)}$.

In order to calculate the length of multiple busy periods, the IF is extended to include more than one condition. The extended IF is denoted as $I_{JST}^S(t)\{(t_s^x, t_e^x, P_r^x)\}$, where $ST$ is the network time at which the extended IF starts, $\{(t_s^x, t_e^x, P_r^x)\}$ is a set of conditions consisting of $(t_0^x, t_0^x, P_0^x), (t_1^x, t_1^x, P_1^x),...,(t_n^x, t_n^x, P_n^x)$. According to the conditions, $I_{JST}^S(t)\{(t_s^x, t_e^x, P_r^x)\}$ is created by considering the frames that arrive in $[t_x^s, t_x^e]$ with a priority higher than $P_r^x (0 \leq x \leq n)$. The $t_x^s, t_x^e$ are relative time to the $ST$.

The exact algorithm using the extended IF is presented in Algorithm 2. For each CIC of $\Gamma_{A(i,m)}$, initialize the start frame $\Gamma_s$, the end frame $\Gamma_e$ and parameter $x$ to $\Gamma_{A(i,0)}$, $\Gamma_{A(i,m)}$ and 0, respectively (lines 02,03). In line 05, search the lowest-priority frame $\Gamma_{A(i,l_x)}$.  

46
Algorithm 2 The Exact WCRT Calculation Algorithm

1: \( R_{A(i,m)} \leftarrow 0 \)
2: for all CICs of \( \Gamma_{A(i,m)} \) do
3: \( \Gamma_s \leftarrow \Gamma_{A(i,0)}; \Gamma_e \leftarrow \Gamma_{A(i,m)}; x \leftarrow 0 \)
4: while \( \Gamma_s \neq \Gamma_e \) do
5: Searching \( \Gamma_{A(i,e)} \) which is the lowest-priority frame queued between \( \Gamma_s \) and \( \Gamma_e \)
6: \( t^s_x \leftarrow \) start of \( \Gamma_s; P^e_x \leftarrow P_{A(i,e)} \)
7: \( t^c_x \leftarrow LCM_{\tau_i \in \Theta\{T_i\}} \)
8: Create \( I^{CIC}J(t) \{ (t^s_0, t^e_0, P^e_0), \ldots, (t^s_1, t^e_1, P^e_1) \} \) for each \( U_J(J \neq I) \), calculate saturation sum of \( I_J(t) \) of all \( U_J \)
9: Calculate sum of \( C_{A(i,0)}; C_{A(i,1)}, \ldots, C_{A(i,e)-1} \)
10: Search the max \( C_k(P_k > P_{A(i,0)}) \) in other stations
11: \( I^{CIC}_J(t) \leftarrow \) Saturation sum results of lines 08-10
12: \( t^e_x \leftarrow EIT(I^{CIC}_J(t)) \)
13: \( \Gamma_s \leftarrow \Gamma_{A(i,e)}; x \leftarrow x + 1 \)
14: end while
15: \( s_{A(i,m)} \leftarrow t^e_x + s_{A(i,0)} \)
16: if \( s_{A(i,m)} + C_{A(i,m)} - a_{A(i,m)} > R_{A(i,m)} \) then
17: \( R_{A(i,m)} \leftarrow s_{A(i,m)} + C_{A(i,m)} - a_{A(i,m)} \)
18: end if
19: end for

between \( \Gamma_s \) and \( \Gamma_e \). Then, initialize the IF conditions \( t^s_x, P^e_x \) to the start of \( \Gamma_s \) and \( P_{A(i,e)} \)
(lines 06). And initialize \( t^e_x \) to a sufficiently big value (i.e., \( LCM_{\tau_i \in \Theta\{T_i\}} \)), so that \( \beta_{A(i,e)-1} \) ends before \( t^e_x \) (line 07).

According to the definition of \( CIC \), calculation of the finish of \( \beta_{A(i,e)-1} \) relates with the following interference time: (1) delay caused by the frames of other stations with higher priority, which will be included in the \( \beta_{A(i,0)}-1, \ldots, \beta_{A(i,e)-1} \) (line 08); (2) delay caused by the earlier queued frames of self station, which can be calculated by the sum of the transmission time of frames \( \Gamma_{A(i,0)}, \ldots, \Gamma_{A(i,e)-1} \) (line 09); (3) delay caused by the frame of other stations with the largest transmission time and a priority lower than \( \Gamma_{A(i,0)} \) (line 10). The saturation addition of these elements is denoted as \( I^{CIC}_J(t) \). Then, the finish of \( \beta_{A(i,e)-1} \) can be calculated by \( EIT(I^{CIC}_J(t)) \). \( EIT(I^{CIC}_J(t)) \) is the operation that finds the first instant at which the slope of the function \( I^{CIC}_J(t) \) becomes 0. Meanwhile, its result is the
Figure 4.7: An example of using IF to calculate response time of $\Gamma_{A(0,1)}$ in its CIC
4.4. THE PROPOSED ALGORITHMS FOR WCRT CALCULATION

CIC is calculated and updated to $R_{A(i,m)}$ in lines 16 and 17. When all candidates have been checked, the maximum response time in all CICs will be the WCRT of the $\Gamma_{A(i,m)}$.

Consider an example to calculate the response time of $\tau_{3,0}$ in the CIC of Figure 4.3. Because $\tau_{3,0}$ is involved in a successive affect frame sequence: $\Gamma_{A(0,0)}(\tau_{7,0})$ and $\Gamma_{A(0,1)}(\tau_{3,0})$, 2 busy periods $\beta_{A(0,0)-1}, \beta_{A(0,1)-1}$ should be considered. First, to calculate the finish of $\beta_{A(0,0)-1}$, the IF condition is initialized to $\{(0, 8, P_7)\}$, since the start of CIC is 0 and the \(\text{LCM}_{\tau_{i} \in \Theta} \{T_i\}\) is 8. Then, the IF of each station is created as shown in Figure 4.7(a). Next, the \(ICIC_{all}\) \{\(0, 8, P_7\)\}(\(t_\)) can be calculated by saturation addition of all \(ICIC_{J}\) \{\(0, 8, P_7\)\} and $C_8$, as shown in Figure 4.7(b). Because $EIT(ICIC_{all}(t))$ is the finish of frames that interfere with $\Gamma_{A(0,0)}$, network time 3 is the end of $\beta_{A(0,0)-1}$, and $\tau_{7,0}$ starts to transmit at this instant.

Since the end of $\beta_{A(0,0)}$ is known, the IF conditions are updated to $\{(0, 3, P_7)(3, 8, P_3)\}$. Then, to calculate the finish of busy period $\beta_{A(0,1)-1}, ICIC_{2}(t)\{(0, 3, P_7)(3, 8, P_3)\}$ and $ICIC_{3}(t)\{(0, 3, P_7)(3, 8, P_3)\}$ are created as shown in Figure 4.7(c). Next, $ICIC_{all}\{(0, 3, P_7)(3, 8, P_3)\}$ is calculated by saturation addition of $ICIC_{2}(t)\{(0, 3, P_7)(3, 8, P_3)\}, ICIC_{3}(t)\{(0, 3, P_7)(3, 8, P_3)\}, C_7,$ and $C_8$, as shown in Figure 4.7(d). Thus, the end of $\beta_{A(0,1)-1}$ is given by the $EIT(ICIC_{all}(t))$, and the response time of $\tau_{3,0}$ (i.e., $\Gamma_{A(0,1)}$) in this CIC is equal to 5, as shown in Figure 4.7(d).

As the same way, the response time of $\tau_{3,0}$ in its other 5 CICs can be calculated. Finally, the maximum response time in the 6 CICs will be the WCRT of $\tau_{3,0}$, which is 7 in this example.

4.4.2 THE APPROXIMATE ALGORITHM

As mentioned before, the exact algorithm has to check every CIC of $\Gamma_{A(i,m)}$. However, the number of CICs will become huge with the increase in the number of messages in a large system, which will result in unaffordable calculation time. Therefore, an approximate algorithm is proposed to speed up the calculation by using the MIF instead of the
Algorithm 3 The Approximate WCRT Calculation Algorithm

1: \( \Gamma_s \leftarrow \Gamma_{A(i,0)} \), \( \Gamma_e \leftarrow \Gamma_{A(i,m)} \), \( x \leftarrow 0 \)
2: \textbf{while} \( \Gamma_s \neq \Gamma_e \) \textbf{do}
3: \quad Searching \( \Gamma_{A(i,L_x)} \) which is the lowest-priority frame queued between \( \Gamma_s \) and \( \Gamma_e \)
4: \quad \( t_s^e \leftarrow \text{start of} \ \Gamma_s, P_s^e \leftarrow P_{A(i,L_x)} \)
5: \quad \( t_e^x \leftarrow \text{LCM}_{\gamma \in \Theta} \{T_i\} \)
6: \quad \textbf{for all} \ \( U_J (J \neq I) \) \textbf{do}
7: \quad \quad Create all \( I_{ST}^J(t) \{ (t_0^s, t_0^e, P_0^e), ..., (t_x^s, t_x^e, P_x^e) \} \) in which \( ST = a_{j,n} \text{ subject to } \phi_J \leq a_{j,n} < \phi_J + \text{LCM}_J, P_J < P_0^e, \tau_J \in U_J \)
8: \quad \quad Create \( M_J(t) \{ (t_0^s, t_0^e, P_0^e), ..., (t_x^s, t_x^e, P_x^e) \} \) as the max of all the \( I_{ST}^J(t) \{ (t_0^s, t_0^e, P_0^e), ..., (t_x^s, t_x^e, P_x^e) \} \) of line 07
9: \quad \textbf{end for}
10: \quad Calculate saturation sum of \( M_J(t) \) of all \( U_J \)
11: \quad Calculate sum of \( C_{A(i,0)}, C_{A(i,1)}, ..., C_{A(i,L_x-1)} \)
12: \quad Calculate \( \max C_k (P_k \geq P_{A(i,0)}, \tau_k \in \Theta) \)
13: \quad \( M_{all}(t) \leftarrow \text{Saturation sum results of lines 10-12} \)
14: \quad \( t_x^e \leftarrow EIT(M_{all}(t)) \)
15: \quad \( \Gamma_s \leftarrow \Gamma_{A(i,L_x)}, x \leftarrow x + 1 \)
16: \textbf{end while}
17: \( s_{A(i,m)} \leftarrow t_x^e + a_{A(i,0)} \)
18: \( R_{A(i,m)} \leftarrow s_{A(i,m)} + C_{A(i,m)} - a_{A(i,m)} \)

IF. MIF based calculation needs only one operation no matter how many CICs exist.

Because MIF is the max of IFs, MIF is also extended to have multiple conditions and denoted as \( M_J(t) \{ (t_n^s, t_n^e, P_n^e) \} \) as done for IF.

The MIF based algorithm for calculation of \( R_{A(i,m)} \) is given in Algorithm 3. In this algorithm, the first step initializes the start frame \( \Gamma_s \), end frame \( \Gamma_e \) and MIF conditions (lines 01-05), which is the same as the exact algorithm. Second, the \( M_J(t) \{ (t_0^s, t_0^e, P_0^e), ..., (t_x^s, t_x^e, P_x^e) \} \) of each station \( U_J (J \neq I) \) is created by line 06-09. The finish of \( \beta_{A(i,L_x)-1} \) is calculated by \( EIT(M_{all}(t)) \) (lines 10-13). Because the \( EIT(M_{all}(t)) \) is the latest finish time of \( \beta_{A(i,L_x)-1} \), the frames that arrive after the \( EIT(M_{all}(t)) \) cannot interfere with \( \Gamma_{A(i,L_x)} \). \( t_x^e \) is thus updated to \( EIT(I_{all}^{CIC}(t)) \) (line 14). For the next while loop, \( \Gamma_s \) is updated to \( \Gamma_{A(i,L_x)} \), and the parameter \( x \) is increased 1 (line 15). The algorithm continues to calculate a new \( t_x^e \) until \( \Gamma_s \) is equal to \( \Gamma_e \). Then, the final \( t_x^e \) will be the latest finish of \( \beta_{A(i,m)-1} \). Finally,
4.4. THE PROPOSED ALGORITHMS FOR WCRT CALCULATION

$EIT(M_{all}(t))$ equals to $t'$

$M_{all}(t)\{(0,8,P_7)\}$

$M_{2}(t)\{(0,8,P_7)\}$

$M_{3}(t)\{(0,8,P_7)\}$

$\Gamma_{A(0,0)}$ $t_{10}$

$\Gamma_{A(0,1)}$ $t_{12}$

Figure 4.8: An example of using MIF to calculate $R_{A(0,1)}$

$R_{A(i,m)}$ can be calculated by using the above results as shown in line 17 and 18.

Consider an example using the message set in Table 4.1. The WCRT of $\tau_{3,0}$ in $U_4$ is calculated as follows. Because $\tau_{3,0}$ is involved in a successive affect frame sequence, 2 busy periods $\beta_{A(0,0)-1}, \beta_{A(0,1)-1}$ should be considered. First, the algorithm creates the MIF for $U_1, U_2, \text{ and } U_3$ with the condition $\{(0,8,P_7)\}$. Note that MIF of $U_1$ is ignored here because it equals zero. Then, $M_{all}(t)\{0,8,P_7\}$ is calculated by saturation addition of $M_2(t)\{(0,8,P_7)\}, M_3(t)\{(0,8,P_7)\}, \text{ and } C_8$. Next, the latest finish of $\beta_{A(0,0)-1}$ is calculated by the $EIT(M_{all}(t)\{0,8,P_7\})$ as shown in Figure 4.8(a). In other words, at this instant, i.e., network time 6, the $\tau_{7,0}$ can be transmitted on the CAN bus. Then, conditions of MIF are updated to $\{(0,6,P_7),(6,8,P_3)\}$ to calculate the next busy period. Finally, the latest finish of $\beta_{A(0,1)-1}$ is calculated by $EIT(M_{all}(t)\{(0,6,P_7),(6,8,P_3)\})$ as shown in Figure 4.8(b). As can be seen, the $\tau_{3,0}$ can be transmitted at network time 7. Therefore, the WCRT of $\tau_{3,0}$ is 7.

From the above example, it is clear that while the IF based exact algorithm needs to check all 6 CICs to calculate the WCRT of $\tau_{3,0}$, the MIF based algorithm only requires
one calculation of $MIF$ for each station. Note that although the results of approximate algorithm are not accuracy, they are equal to or larger than the real WCRT in all cases according to the feature of $MIF$ operation. Therefore its results are sufficiently safe from the WCRT analysis point of view.

### 4.4.3 Computational Complexity Analysis

To analyze the computational complexity of the proposed two algorithms, let us assume that $n$ stations exist in the network, $\Gamma_{A(i,m)}$ is a frame of $U_I$, which belongs to $\Gamma_{A_i-seq}$. Also, assume that each station $U_J$ has $N_J$ frames with priorities higher than $P_{A(i,l_0)}$. The $P_{A(i,l_0)}$ is the priority of $\Gamma_{A(i,l_0)}$ which is the lowest-priority frame between $\Gamma_{A(i,0)}$ and $\Gamma_{A(i,m)}$. The number of CICs of $\Gamma_{A(i,m)}$, which represents the computational complexity of the exact algorithm, is as follows:

$$\text{Complexity}_{\text{exact}} = \prod_{J \in \Theta, J \neq I} N_J$$  \hspace{1cm} (4.1)

However, the computational complexity of the approximate algorithm is given as follows:

$$\text{Complexity}_{\text{approximate}} = \sum_{J \in \Theta, J \neq I} N_J$$  \hspace{1cm} (4.2)

From the above equations, it is clear that the approximate algorithm can extremely decrease the computational complexity in a large system.

### 4.5 Evaluation

In order to validate the efficiency of the proposed methods, experiments are conducted by using message sets generated by NETCARBENCH [26] and a real message set provided
by an automaker. All the experiments are performed on a computer with an Intel Core i5 2.67GHz processor. As mentioned in the Section 2.3, all stations are assumed to use the FIFO queue.

4.5.1 Experiments on NetCarBench-generated Message Sets

In the first experiment, WCRT of message is analyzed on 10 small message sets generated by NetCarBench. All the message sets are configured as a typical 500 kbps powertrain network with a bus load of 20%-25%, station number 3-5 and message number 50-76. Deadlines of messages are assumed to be equal to their periods. Priority of messages are assigned based on the deadline monotonic algorithm: the shorter deadline message has, the higher priority message is assigned. The offset of each message is assigned by NetCarBench [26].

Results of experiment 1 are shown in Table 4.2. While the exact algorithm takes an average time of 492.82 seconds to finish the calculation, the approximate algorithm only needs an average time of 1.02 seconds. As for accuracy, the approximate algorithm has an average of 7.66% messages with different results from the the exact algorithm. Specifically, although two message sets have not error, the approximate algorithm achieves an average error of 1.95%, and a max error of 8.3% comparing with the exact algorithm. The error of a message $\tau_i$ is calculated by the following formula:

$$\text{Error}_i = \left( \frac{R_{i}^{\text{approximate}} - R_{i}^{\text{exact}}}{R_{i}^{\text{exact}}} \right) \times 100\%$$

(4.3)

where $R_{i}^{\text{approximate}}$ and $R_{i}^{\text{exact}}$ are the WCRT of $\tau_i$ calculated by approximate algorithm and exact algorithm, respectively.
Table 4.2: Experiment 4.1: Error and run time comparisons between the exact and approximate algorithms based on 10 message sets generated by NETCARBENCH

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Message with error</th>
<th>Max Error</th>
<th>Average Error</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>492.82 s</td>
</tr>
<tr>
<td>Approximate</td>
<td>7.66%</td>
<td>8.3%</td>
<td>1.95%</td>
<td>1.02 s</td>
</tr>
</tbody>
</table>

4.5.2 Experiments on an Automaker-provided Message Set

To validate the efficiency of the proposed algorithms on a real system, the message set provided by an automaker is used, which composes of 14 stations and 66 messages, and has 53.3% bus load. All message properties, including the offset, were configured by the automaker. Deadlines of messages are configured to equal to their periods. Priority of messages are assigned mainly based on the deadline monotonic algorithm. Offset assignment of the messages are similar to the NETCARBENCH method.

Because the system is too large to check all stations by the exact algorithm, one station of this network system is tested, which includes 15 messages. As shown in Table 4.3, while the exact algorithm required 7 days to finish the analysis of WCRT for 15 messages, the approximate algorithm only required 17.46 seconds. Note that although there are no errors on the 15 messages, it does not mean that the approximate algorithm can always obtain the same results as the exact algorithm as indicated in the first experiment.

Table 4.3: Experiment 4.2: Error and run time comparisons between the exact and approximate algorithms based on a real message set

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Message with error</th>
<th>Max Error</th>
<th>Average Error</th>
<th>Run time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7 days</td>
</tr>
<tr>
<td>Approximate</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>17.46 s</td>
</tr>
</tbody>
</table>
4.6. EXTENSION WITH CONSIDERATION OF JITTER

Table 4.4: Experiment 4.3: A WCRT comparison of a real message set with and without offset

<table>
<thead>
<tr>
<th>Decrease rate</th>
<th>Message Average WCRT</th>
<th>Maximum WCRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>96.97%</td>
<td>42.56%</td>
<td>64.21%</td>
</tr>
</tbody>
</table>

4.5.3 EXPERIMENTS ABOUT EFFECTIVENESS OF OFFSET ASSIGNMENT

The same message set, which is used in the second experiment, is employed to confirm the effectiveness of assigning offset to messages in a FIFO queue system, Table 4.4 illustrates the decrease rate of average WCRT and maximum WCRT of the messages set after assigning offset. As can be seen, there are 96% messages’ WCRT are decreased after being assigned offset. In these messages, average decrease rate of WCRT is 42%, maximum decrease rate of WCRT is 64%, The results confirmed that assigning offset can also extremity decrease WCRT of messages in the FIFO queue system.

4.6 EXTENSION WITH CONSIDERATION OF JITTER

As mentioned before, it is assumed that there is no jitter on the arrivals of the messages for the purpose of simplifying analysis. However, considering that the existence of jitter can affect WCRT of messages, the extension with consideration of jitter is explained in this section.

4.6.1 EFFECTS OF JITTER ON THE PROPOSED METHOD

For message \( \tau_i \), the queueing process takes a bounded amount of time, between 0 and \( J_i \), before \( \tau_i \) is queued available for transmission. \( J_i \) is referred to as the maximum queuing jitter of \( \tau_i \). Considering jitter, frame \( \tau_{i,m} \) may arrives at any time between \([a_{i,m}, a_{i,m} + J_i]\).

The occurrence of jitter leads arrival times and queuing order of messages to become
changeable, which betrays assumptions of the proposed method. The proposed definitions, successive frame sequence and successive affect frame sequence, thus are not suitable for the jitter model. However, this problem can be solved by extending the proposed definitions.

4.6.2 Extended Successive Affect Frame Sequences

Because \( J_i \) of each \( \tau_i \) is fixed when message set is generated, all the possible queuing orders of messages in each station are finite and can be analyzed. Denote successive frame sequences of messages in same station as \((\Gamma^k_0, \ldots \Gamma^k_n)(k = 0, 1, \ldots)\). Each sequence is referred as a queuing order of messages in this station. For example, assume \( \tau_{1,0}, \tau_{2,0}, \tau_{3,0} \) are frames of same station, arrival time and maximum jitters of these messages are shown in Figure 4.9. It is clear that there are 2 queuing orders of messages in this station: \((\tau_{1,0}(\Gamma^k_0), \tau_{2,0}(\Gamma^k_1), \tau_{3,0}(\Gamma^k_2))\) and \((\tau_{2,0}(\Gamma^k_0), \tau_{1,0}(\Gamma^k_1), \tau_{3,0}(\Gamma^k_2))\).

In each successive frame sequence \((\Gamma^k_0, \ldots \Gamma^k_n)\), the successive affect frame sequence of \( \Gamma^k_i \), denoted as \( \Gamma^k_{A_i-seq} \), can be found by the definition of Effect in Section 4.1.

4.6.3 Worst-case Jitter Occurred CICs of \( \Gamma^k_{A(i,m)} \)

Based on the extended successive affect frame sequences, CICs of \( \Gamma^k_{A(i,m)} \) can be found by considering all the queuing orders of messages in other stations. In each CIC of \( \Gamma^k_{A(i,m)} \),

![Figure 4.9: An example showing effects of jitter on the queuing order of messages](image)
because queuing order of messages is unique, WCRT of $\Gamma_{A(i,m)}^k$ will be largest if frames arriving at the CIC have the jitter as large as possible, and the frames arriving after the CIC have the jitter as small as possible. ¹ This kind of CIC's is defined as the worst-case jitter occurred CICs of $\Gamma_{A(i,m)}^k$. Because queuing order and arrival times of messages are fixed in each worst-case jitter occurred CIC of $\Gamma_{A(i,m)}^k$, WCRT of $\Gamma_{A(i,m)}^k$ thus can be calculated by the proposed algorithms.

However, the arriving order of messages may have many situations if the maximum jitter of messages are large. In this case, finding the worst case arriving order firstly then calculating WCRT of $\Gamma_{A(i,m)}^k$ will be a solution for speeding up the calculation. This will be considered as future work.

4.7 SUMMARY

In this chapter, a WCRT analysis method for messages in the FIFO-based and offset assigned CAN systems has been proposed. First, a critical instant theorem has been given and proved. Then, two algorithms for calculating the WCRT based on the given theorem have been proposed. The exact algorithm can obtain accurate results with large computational cost, which is suitable for small system. In contrast, the approximate algorithm can analyze a larger system with limited errors and low computational complexity. Experimental results on generated message sets and a real message set have validated the effectiveness of the proposed two algorithms. Also, experiment on a real message set showed assigning offset to the messages can decrease the WCRT significantly in the FIFO queued system.

¹Similar conclusion about the worst-case jitter has been summarized by [4, 5], [10, 19].
CHAPTER 5

SCHEDULABILITY COMPARISON BETWEEN PRIORITY QUEUES AND FIFO QUEUES FOR CAN MESSAGES WITH OFFSETS

5.1 OVERVIEW

In previous research, the schedulability analysis for CAN messages with offset in a system using priority queues or FIFO queues — the two most popular queues in real-world CAN stations — have been proposed independently. However, a schedulability comparison between priority queues and FIFO queues for CAN messages with offsets was never conducted. In order to find the pros and cons of the two types of queues, the schedulability comparison of the two types of queues is conducted in this chapter. Based on the results, a new P&F scheduling method, which uses both priority and FIFO queues in a single station, is proposed to combine the advantages of the two types of queues.
This chapter is organized as follows. Section 5.2 gives the comparing experiments and shows the pros and cons of priority and FIFO queues. Based on the results, Section 5.3 proposes a P&F queue based scheduling method that uses both priority and FIFO (P&F) queues in a single CAN station. Section 5.4 evaluates the P&F scheduling method by using a real-world message set provided by the automobile manufacturer. Finally, this chapter is summarized in Section 5.5.

5.2 Schedulability Comparison

Deadline Rate and Worst-case Response Time Rate, introduced in Section 3.2, are also used for the schedulability comparison in this chapter. Based on these definitions, the schedulability of a system using a queue A is higher than the one using another queue B if the maximum $RR_i$ of messages in queue A is lower than that in queue B.

5.2.1 Experiments on Normal Message Sets

Two message sets are employed for the comparison. The message sets are generated by NETCARBENCH and configured as typical 500 kbps powertrain network with 31.3% bus load, 42 messages and 54.8% bus load, 58 messages. Offset is assigned for each message set by NETCARBENCH. Deadlines of messages are assumed to equal to their periods. Priority of messages are assigned based on the deadline monotonic — the message with shorter deadline gets higher priority.

Results of the comparisons are shown in Figure 5.1. In the figures, the red line indicates the result of the priority queue, which is calculated by assuming all stations in the system use the priority queue. The blue line indicates the result of the FIFO queue, which is calculated by assuming all stations use the FIFO queue.

First, focus on the WCRT results shown in Figure 5.1(a) and Figure 5.1(b). It can
5.2. SCHEDULABILITY COMPARISON

Figure 5.1: WCRT and WCRR results of message sets generated by NETCARBENCH. The two message sets are configured as a typical 500kbps powertrain network with 31.3% bus load, 42 messages and 54.8% bus load, 58 messages.

be found from the results that the high-priority messages in priority queue have shorter WCRTs than those in FIFO queue. However, the low-priority messages have larger WCRTs in the priority queue. With the decrease of message priority (increase of the value of priority), the WCRTs of messages in the priority queue increase fast. In contrast, in the FIFO queue, the WCRTs of the high-priority and the low-priority messages are closer than those in the priority queue. With the decrease of message priority, the WCRTs of messages in the FIFO queue increase more slowly than those in the priority queue. Also, the max WCRT of messages in the FIFO queue is shorter than that in the priority queue. These phenomena exist in the results of both message sets, but are more obvious with the increase of the bus
Then, the WCRR results of messages are shown in Figure 5.1(c) and Figure 5.1(d). In the figures, the max WCRR of messages in the priority queue is smaller than that in the FIFO queue. Also, the difference of the max WCRR becomes larger with the increase of the bus load. Thus, for these two message sets, the schedulability of the priority queue is higher than that of the FIFO queue.

5.2.2 EXPERIMENTS ON SPECIAL MESSAGE SETS

In the priority queue, low-priority message will be delayed by high-priority messages that are queued before or later, so that ensures the transmission of high priority message. In contrast, priority inversion occurs in FIFO queue and cause high priority message experiencing more delays in the worst-case, but it avoids low priority message being delayed successively. Thus, a part of delays of low priority message is shared by high priority message, so that the difference of WCRT between high and low priority messages becomes close. These explain the results of Figure 5.1. With the decrease of priority, WCRT of FIFO queued message increase slower than it of priority queued message. The max WCRT is smaller in FIFO queued message than in priority queued message.

Because schedulability is evaluated by WCRT and period, neither priority nor FIFO queue can always owns higher schedulability than another. The schedulability of a message set is depended on both scheduling method and feature of the message set. For example, the priority queue will have high schedulability if the period of high-priority message is much shorter. In contrast, if the period of low priority message is equal to the period of high priority message, the FIFO queue will be better.

Summarizing the above, the conclusion can be reached that in a system, in which each station only includes messages with the same period, the schedulability of the FIFO queue will be higher than that of the priority queue. To verify this conclusion, two special message
5.3. SCHEDULING MESSAGES WITH BOTH FIFO AND PRIORITY QUEUES

Figure 5.2: WCRR results of message sets generated by NETCARBENCH. Each station is configured to only include messages with the same period. The properties of the two message sets are 52.5% bus load, 79 messages, and 79.9% bus load, 92 messages.

sets were generated based on the configuration of the 500kbps powertrain network, but each station only included messages with the same period. The properties of the message sets were 52.5% bus load, 79 messages, and 79.9% bus load, 92 messages. Also, an offset was assigned for each message.

The WCRR results of the message sets are shown in Figure 5.2. In the lower bus load situation shown in Figure 5.2(a), the max WCRR of messages is smaller in the FIFO queue than in the priority queue. Furthermore, in the higher bus load situation shown in Figure 5.2(b), the FIFO queue system shows more advantages.

5.3 SCHEDULING MESSAGES WITH BOTH FIFO AND PRIORITY QUEUES

Although the FIFO queue shows an advantage in the experiments of Figure 5.2, the situation that each station only includes messages with the same period hardly exists in a real automotive system. To combine the advantages of both the priority and the FIFO queue in a real system, a new scheduling method — called P&F scheduling method — is proposed.
The P&F scheduling method is assumed to be employed in a station that includes both the priority queue and the FIFO queue. The arbitration of different queues in the station is based on priority: when messages of different queues attempt to transmit to the CAN-bus at the same time, the higher-priority message will be transmitted first. Note that the P&F scheduling method is reasonable because some CAN-controller embedded commercial MCUs equip both hardware priority queue and FIFO queue. (e.g., the R32C/160 group MCU from Renesas [25]).

An example is given in Figure 5.3 to describe the transmission of messages in the P&F station. In this example, the station has 4 messages. ID:1 and ID:2 messages are put in the FIFO queue, and ID:3 and ID:4 messages are put in the priority queue. When the CAN controller attempts to transmit the messages, the earliest queued message (ID:2) of the FIFO queue arbitrates with the highest-priority message of the priority queue (ID:3). Then the ID:2 message is selected by the CAN controller and attempts transmission to the CAN-bus.

The main idea of the P&F scheduling method is to find the message with the max WCRR in the priority queue, then schedule the messages that have same period with the
5.4. EVALUATION OF THE P&F SCHEDULING METHOD

To evaluate the P&F scheduling method, a schedulability comparison between the priority queue, the FIFO queue and the P&F queue is conducted by using a real-world message set provided by the automobile manufacturer. The message set has 53.3% bus load and 65 messages. All message properties, including the offset, were configured by the manufacturer.

max WCRR message to FIFO queue, so that the max WCRR can be decreased. Meanwhile, the other messages having a large WCRT are put into the priority queue so that they do not delay the FIFO-queued messages. The details of this method are given as follows:

1. Assume all stations use priority queue, calculate WCRT of all messages.

2. For each $U_i$, find the $\tau_i (\tau_i \in U_i)$ with max WCRR. Then, put the messages of $U_i$ which has same period with $\tau_i$ into FIFO queue. The other messages of $U_i$ are put into priority queue.

Figure 5.4: WCRR results of the message set provided by the automobile manufacturer. The message set has 53.3% bus load and 65 messages. All message properties, including the offset, were configured by the manufacturer.
The results are shown in Figure 5.4. First, a schedulability comparison between priority queue and FIFO queue is shown in Figure 5.4(a). The performance of priority queue is better than that of FIFO queue. Then, the comparison between priority queue and P&F is shown in 5.4(b). For the P&F used situation, WCRR of messages in the priority queue of P&F is calculated by the method of Section 2.5. WCRR of messages in the FIFO queue of P&F is calculated by the approach proposed in Chapter 4. As can be seen, the max WCRR of the message set decreased. This result confirms that the schedulability of the system is improved by the P&F scheduling method.

5.5 SUMMARY

In this chapter, a schedulability comparison between priority queues and FIFO queues for CAN messages with offsets is conducted. The results concluded that the priority queue can ensure that high-priority messages have a short WCRT, but the FIFO queue can decrease the max WCRT of messages in the queue. For the result of schedulability, the priority queue achieves better results for normal message sets, but the FIFO queue achieves better results for special message sets, in which each station only includes messages with the same period. Considering the comparison results, the new P&F scheduling method that uses both priority and FIFO queues in one CAN station is proposed. The experimental results of a real message set from an automobile manufacturer show that the combination of priority and FIFO queues can achieve higher schedulability than either the priority or the FIFO queue alone.
CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

6.1 CONCLUSIONS

In recent years, the sophistication and complexity of in-vehicle networks has seen a notable increase in the number of transmitted messages. As a result, guaranteeing the deadlines for all messages in the network has become a difficult challenge. In this context, scheduling messages with offsets has emerged as an effective method to improve the schedulability of modern CAN networks. For the purpose of increasing the utility of the offset-based message scheduling, so that to improve the schedulability for the modern CAN networks, an offset assignment method; a schedulability analysis method for CAN messages with offsets in a system using FIFO queues; and a schedulability comparison between priority queues and FIFO queues for CAN messages with offsets have been proposed in this dissertation.

The proposed offset assignment method employs the Simulated Annealing (SA) algorithm for a better offset assignment with respect to the WCRT. The method initializes offsets by restricting the objective function of SA, which is selected as Maximum Interference Function obtained from messages in each station, as small as possible. Then, based on the WCRT corresponding to the initial offset, the method modifies the ob-
jective function and reforms the offset assignment to guarantee that all messages meet their deadlines. Evaluation of the proposed method on 4 message sets showed that the proposed method can decrease average $RR_i$ by 4% to 33%, maximum $RR_i$ by 3% to 35%, comparing with the previous method. Specially, in the situation that the message set has high bus load of messages in each station, the proposed method shows more advantages. These results confirmed that the proposed method can effectively improve the schedulability of CAN messages. Although calculation time of the proposed method is more than that of the previous method (In the experiments which are shown in Section 3.5, the proposed method finished the calculation in 30 minutes, the previous method finished the calculation in 5 minutes), a calculation time of 30 minutes is acceptable for a real world offset assignment application.

In the proposed schedulability analysis method for CAN messages with offsets in a system based on FIFO queues, first, a new critical instant theorem has been proposed to locate the worst case situation for a given message. Then, based on this theorem two algorithms for calculating the WCRT have been proposed — an exact algorithm for accurate calculation; and an approximate algorithm for rapid estimation. The exact algorithm is computationally demanding, and is suitable for evaluating the results of the approximate algorithm or for calculations in small systems. In contrast, the approximate algorithm can estimate the WCRT with limited errors while having a low computational complexity; and it is useful for larger systems. It has been proved that the exact algorithm is accurate and the approximate algorithm is sufficient safety for WCRT calculation. Furthermore, experiments have been conducted to validate the above two algorithms by using synthetically generated message sets and a real-world message set from an automobile manufacturer. The results confirmed the accuracy and efficiency of the proposed method.

Finally, a schedulability comparison between priority queues and FIFO queues for CAN messages with offset has been conducted. The experimental results show that priority
queues can ensure that high-priority messages have a short WCRT, while FIFO queues can decrease the max WCRT of messages in the queue. For the result of schedulability, priority queues achieve better results for normal message sets, but FIFO queues achieve better results for special message sets, in which each station only includes messages with the same period. Considering the comparison results, a new P&F scheduling method that uses both priority and FIFO queues in a single CAN station has been proposed. The experimental results using a real message set from an automobile manufacturer show that the combination of priority and FIFO queues can achieve higher schedulability than either priority or the FIFO queues alone.

6.2 FUTURE WORKS

In this dissertation, the jitter has not been considered into the proposed methods for the purpose of simplifying the complexity of analysis, although it is a practical issue. However, in order to improve the practical application of offset-based message scheduling, it is important to take queuing jitter into account, and it will be one of the future work.

To cope with the increasing transmission data, in-vehicle network with more complex topology has been developed. As a practical solution, multiple CAN network can be integrated via gateways. However, related analysis and theory have not been developed yet. Therefore, extension of the schedulability analysis theory to consider effects of the gateways will also be the future work.
BIBLIOGRAPHY


BIBLIOGRAPHY


http://www.renesas.com/products/mpumcu/m16c/m16c50/Documentation.jsp
http://inst.eecs.berkeley.edu/ee249/fa08/Lectures/handout_canbus2.pdf.


ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to my supervisor Professor Hiroaki Takada. He taught me the way of solving problems and let me understand the fun of thinking.

I deeply indebted to Associate Professor Gang Zeng for giving invaluable suggestions, directions and kindly encouragement throughout my studies.

I own a special debt of appreciate to Professor Hiroyuki Tomiyama for offering me an education opportunity in Lab Takada and giving me invaluable suggestions on my studies. I also sincerely express my respect and appreciation to Ph.D. Ryo Kurachi for his suggestions and valuable technical supports.

I would like to express my special thank to Professor Shoji Yuen for his insightful comments and advice on my studies. Also, I would like to thank all the other members in Lab Takada for their help in the completion of this thesis.

Finally, I deeply thank my family for their helping and supporting throughout my studies.
LIST OF PUBLICATIONS BY THE AUTHOR

RESEARCH ACHIEVEMENTS RELATED TO THE DISSERTATION

JOURNAL PAPERS


INTERNATIONAL CONFERENCES

LIST OF PUBLICATIONS BY THE AUTHOR

OTHER ACHIEVEMENTS