Worst Case Response Time Analysis for Messages in Gateway-Interconnected Controller Area Network

YONG XIE[†1†3] GANG ZENG[†2] YANG CHEN[†3]
RYO KURACHI[†3] HIROAKI TAKADA[†3] RENFA LI[†1]

Abstract: Controller Area Network (CAN) is widely used inside the automobiles. To decrease design complexity and cost, gateway is employed to realize the communication between different CAN buses. But its employment brings great challenges for worst-case response time (WCRT) analysis of CAN messages. We first analyzed the key challenges for WCRT analysis of messages. And then, based on existing method proposed for one single CAN, a new WCRT analysis method that considers the timing distance relations among messages is proposed for non-gateway messages. Furthermore, a division-based method that transforms the end to end WCRT analysis of gateway messages into the similar case with WCRT analysis of non-gateway messages is proposed for gateway messages. The correctness of the proposed method is proved and its usability is verified by comparing it with a full space searching based simulator as well.

Keywords: CAN, Worst Case Response Time, Gateway, Busy Sequence, The Minimum Distance Constraint

1. Introduction

To meet the requirements from safety, energy efficiency and infotainment, more and more sensors, actuators and ECUs are added into the automotive electronic system, which increased the complexity of the automotive networks to a large extent [1]. CAN is currently the most widely used communication technology inside the automotive electronic system. To reduce design complexity and cost, several CAN buses are utilized in different sub systems, such as the body system, powertrain system and information system. Therefore, gateway is employed to enable the communication between different CAN buses [1, 2], and the sophisticated functions such as collision detection, vehicle dynamic integrated management and pre-crash safety are realized based on the message exchange through the gateway. Fig 1 shows an example of gateway-based implementation of the collision detection system [3]. The basic function of gateway is to realize the message exchange between different CAN buses, where messages from one CAN are first stored in queue inside the gateway and then forwarded into another CAN when they win the arbitration. But other complex functions such as jitter reduction and message filtering, which can reduce the worst case response time (WCRT) for messages and the bus load for CAN buses can also be implemented inside the gateway [4, 5].

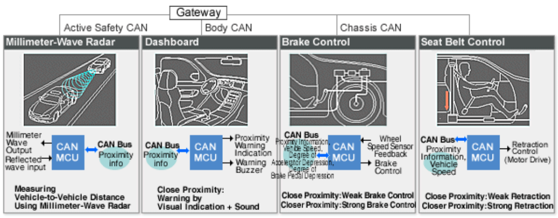


Fig 1: Gateway-based implementation of Collision Detection System [3]

As automotive electronic system is generally a hard real-time system, we must guarantee the real-time capability of CAN messages, or else it will result in a catastrophic situation. Thus, we must analyze the WCRT for CAN messages. Although the message's WCRT analysis for one single CAN has attracted much attention since 1994[6, 7], the adding of gateway brings many new challenges, thus the existing methods that were proposed for one single CAN cannot be reused. There are very few related works about the gateway-interconnected CAN buses: Sommer [4] proposed a CAN-CAN gateway embedded system, where the resource dimensioning problem such as the gateway processing time and the buffer capacity are investigated; Davis [5] proposed a method to reduce the jitter for gateway messages; Sojka [8] proposed a measurement-based method to analyze the latency introduced by gateway. But no WCRT analysis method is proposed for messages in these works. Although there are a few industry tools that claiming to support the message's WCRT analysis for gateway-interconnected CAN buses, their method is confidential. This is the first time, to the best of our knowledge, that a complete and detailed approach is presented and proven.

The main contributions of this work are as follows: first, it analyzed the main challenges for WCRT analysis of messages in gateway-interconnected CAN buses; and then it proposed an analysis method to tackle such challenges, the correctness of the proposed method is proved and the usability of it is verified through comparing with a full space searching based simulator.

2. System Model and Key Assumptions

As shown in Fig 2, we assume that automotive electronic system consists of two sub systems, and each sub system includes several ECUs that connected by a CAN bus. The included two CAN buses have the same bandwidth and are interconnected by gateway, where messages are exchanged to realize the communication between two sub systems. Each ECU ECU_N contains a message set S_N that needs to be transmitted on CAN. As the priority is unique for each message for the whole system, we use the symbol m_i with the subscript representing its priority to indicate it. Thus, $S_N = \{m_i | m_i \in S_N\}$. m_i is indicated by a 4-tuple: $< T_i , P_i , C_i , D_i >$, which represent the period, priority, transmission time and deadline, respectively, and we assume

†1 Laboratory of Embedded Systems and Networking, Hunan University.
†2 Graduate School of Engineering, Nagoya University
†3 Graduate School of Information Science, Nagoya University

that $T_i = D_i$. If $i < j$, it means that m_i has higher priority than m_j. For two communicating ECUs that belong to the same sub system, the communicating messages will be only transmitted on the included CAN of the corresponding sub system, we define this kind of message as non-gateway message and the included CAN bus as its source CAN CAN_{sou}. Such as m_2 in ECU_1 is a non-gateway message and its CAN_{sou} is CAN_1. For two communicating ECUs that belong to different sub systems, the communicating messages will be transmitted on its CAN_{sou} first, and then go through the gateway and be transmitted on the other CAN again, we define this kind of message as gateway message and the other CAN as its destination CAN CAN_{des}. Such as m_1 in ECU_1 is a gateway message, its CAN_{sou} is CAN_1 and its CAN_{des} is CAN_2.

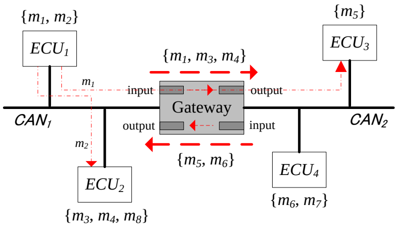


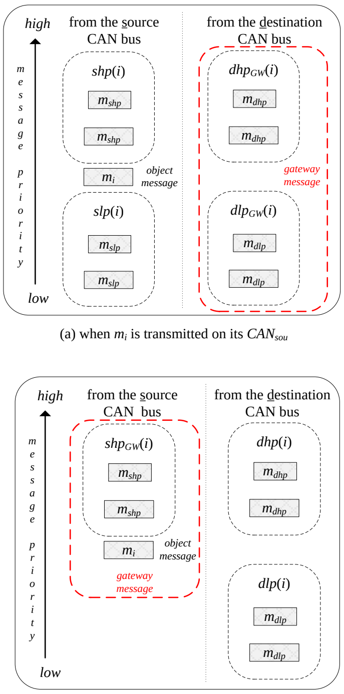
Fig 2: System architecture for automotive electronic system

We also make the following assumptions about the gateway: in each transmission direction for example from CAN_1 to CAN_2, there is a set of queues (include the input and output queue) to realize the store and forward operation for message exchange, hence the messages from different transmission directions will not interfere with each other inside the gateway; we also assume that queues' size are limitless and they are managed with the fixed-priority based policy, and we ignore the gateway processing time for gateway messages.

3. Problem Analysis

For the following parts of this paper, we always assume that the object message for WCRT analysis is m_i. We define the WCRT of m_i as the maximal interval between the release time in its host ECU and the arrival time in its destination ECU. However, as the transmission path for non-gateway and gateway message is different, we have to differentiate the WCRT for those two kinds of messages. If m_i is a non-gateway message, it will be transmitted on its CAN_{sou} only, thus we indicate its WCRT as $r_{s,i}$. But if m_i is a gateway message, the transmission path will includes its CAN_{sou}, gateway and its CAN_{des}, the corresponding WCRT is usually called end to end WCRT, thus we indicate it as $r_{e2e,i}$.

For WCRT analysis of m_i, the key is to analyze all the possible interference that would happen to it, thus next we will focus on interference analysis for m_i. If m_i is a non-gateway message, four different types of messages will interfere with m_i as shown in Fig 3(a). $shp(i)$ and $slp(i)$ represent two sets of messages that come from m_i's CAN_{sou} and have higher and lower priority than m_i, respectively. $dhp_{GW}(i)$ and $dlp_{GW}(i)$ represent another two sets of gateway messages that come from m_i's CAN_{des} and have higher and lower priority than m_i, respectively. The subscript GW indicates the message set is a gateway message set. $dhp_{GW}(i)$ is only a sub set of $dhp(i)$, $dhp(i)$ also include other non-gateway messages. The similar situation also happens to $dlp_{GW}(i)$. For example for m_2 in Fig 2, $shp(2)=\{m_1\}$, $slp(2)=\{m_3, m_4, m_8\}$, $dhp(2)=\{\}$, $dhp_{GW}(2)=\{\}$, $dlp(2)=\{m_5, m_6, m_7\}$, $dlp_{GW}(2)=\{m_5, m_6\}$. As $shp(i)$ messages belong to the same CAN bus with m_i, thus their interference pattern is periodic and we can reuse the existing method proposed for one single CAN to analyze its interference[7]. For both $slp(i)$ and $dlp_{GW}(i)$ messages, as they can cause the priority inversion to m_i only once, we can include its interference by choosing the message with the maximal C_k as [7] did. But for $dhp_{GW}(i)$ messages, they need to be scheduled in m_i's CAN_{des} first, and then arrive at m_i's CAN_{sou} and cause interference on m_i. The finishing time in m_i's CAN_{des} is equal to its arriving time in m_i's CAN_{sou} for $dhp_{GW}(i)$ messages because of the ignoring of gateway processing time. As the response time for each instance of $dhp_{GW}(i)$ messages in m_i's CAN_{des} is different, the arriving pattern of $dhp_{GW}(i)$ messages in m_i's CAN_{sou} is dynamic. Thus, how to define the interference from the dynamic arriving $dhp_{GW}(i)$ messages is a challenge for $r_{s,i}$ analysis of m_i. Furthermore, as shown in Fig 2, the gateway messages from CAN_1 such as $\{m_1, m_3, m_4\}$ will interfere with the messages that belong to CAN_2 such as $\{m_5, m_6, m_7\}$ in CAN_2, and conversely the gateway messages from CAN_2 such as $\{m_5, m_6\}$ will also interfere with the messages that belong to CAN_1 such as $\{m_1, m_2, m_3, m_4, m_8\}$ in CAN_1. Consequently, for $r_{s,i}$ analysis of m_i, another challenge that is also brought by gateway messages is the inter-dependency between messages in two CAN buses.



(a) when m_i is transmitted on its CAN_{sou}

(b) when m_i is transmitted on its CAN_{des}

Fig 3: Interference analysis for CAN messages

If m_i is a gateway message, the complexity for $r_{e2e,i}$ analysis will be much bigger compared with that of $r_{s,i}$, because m_i will be scheduled in two CAN buses, thus we need to analyze the interference that would happen to m_i in both of these two CAN buses. When m_i is scheduled in its CAN_{sou}, all the possible kinds of interference is the same with that for $r_{s,i}$ analysis of non-gateway messages as shown in Fig 3(a). When m_i is scheduled in its CAN_{des}, another three types of messages will interfere with m_i as shown in Fig 3(b). $dhp(i)$ and $dlp(i)$ represent two sets of messages that belong to m_i's CAN_{des} and have higher or lower priority than m_i, respectively. $shp_{GW}(i)$ represents the set of gateway messages that belongs to m_i's CAN_{sou} and has higher priority than m_i. There will also be $slp_{GW}(i)$ messages that belong to m_i's CAN_{sou} and has lower priority than m_i, but as it cannot arrive at m_i's CAN_{des} at the same time with m_i (the reason will be explained in Section 4), thus it cannot cause the priority inversion to m_i. Take m_3 in Fig 2 for example, when it is transmitted on its CAN_{sou}, $shp(3)=\{m_1, m_2\}$, $slp(3)=\{m_4, m_8\}$, $dhp_{GW}(3)=\{\}$, $dlp_{GW}(3)=\{m_5, m_6\}$; when it is transmitted on its CAN_{des}, $shp_{GW}(3)=\{m_1\}$, $slp_{GW}(3)=\{m_4\}$, $dhp(3)=\{\}$, $dlp(3)=\{m_5, m_6, m_7\}$.

For both $dlp_{GW}(i)$ and $slp(i)$ messages when m_i is transmitted on its CAN_{sou} and $dlp(i)$ messages when m_i is transmitted on its CAN_{des}, as they can cause priority inversion to m_i only once, we can easily include their interference by choosing the message with the maximal C_k separately. When m_i is transmitted on its CAN_{sou}, the interference pattern of $shp(i)$ messages is periodic, but part of $shp(i)$ messages are also gateway messages as well, which are the $shp_{GW}(i)$ messages when m_i is transmitted on its CAN_{des}, thus their interference pattern is changed to be dynamic. And the interference caused by $shp_{GW}(i)$ messages in m_i's CAN_{des} and m_i's CAN_{sou} is inter-dependent. The same situation also happens to $dhp(i)$ messages when m_i is transmitted on its CAN_{des}. For example take m_6 in Fig 2 for example, when m_6 is transmitted on its CAN_{sou} CAN_2, m_5 belongs to $shp(i)$ and its interference pattern is periodic, m_1 belongs to $dhp_{GW}(i)$ and its interference pattern is dynamic; but after m_6 going through the gateway and arriving at its CAN_{des} CAN_1, m_5 belongs to $shp_{GW}(i)$ and its interference pattern is changed to be dynamic, m_1 belongs to $dhp(i)$ and its interference pattern is changed to be periodic. Therefore, the challenges that are met for $r_{s,i}$ analysis of non-gateway messages are also exist for $r_{e2e,i}$ analysis. Furthermore, as the interference caused by higher priority gateway messages is inter-dependent in two CAN buses and the interference pattern is changeable from periodic to dynamic or conversely for the same gateway message, to define in what kind of situation the higher priority gateway messages will cause the maximal interference on m_i from the end-to-end's point of view is another much bigger challenge for $r_{e2e,i}$ analysis.

4. Interference Analysis for Gateway Messages

After the above analysis, we can find that the key challenges for WCRT analysis of messages in gateway-interconnected CAN buses are all coming from the gateway messages. As a result, we will focus on how to analyze the interference from gateway messages in this section. To clarify the description, we assume that m_i is a non-gateway message, and the analysis object are $dhp_{GW}(i)$ messages that come from its CAN_{des} and has priority higher than m_i. m_k belongs to $dhp_{GW}(i)$, and we will show how to define m_k's interference on m_i. As explained before, the arriving pattern of m_k in m_i's CAN_{sou} is dynamic, thus one classical approach is to treat it as a sporadic message and set the closest distance between the arriving time of two continuous instances of m_k in m_i's CAN_{sou} as its period just like [5] did. By doing this, the dynamic arriving m_k is transformed into a periodic message in m_i's CAN_{sou} and we can reuse the existing method to analyze its interference [7]. But this approach will bring much pessimism to the interference analysis of m_k, as the response time of its instances is changeable between C_k and $r_{s,k}$, thus the variation range of the distance between the arriving time of two continuous instances of m_k in m_i's CAN_{sou} is very large. Therefore, we propose a new definition *busy sequence* to capture the characteristic of dynamic arriving m_k, which can get a tighter bound of its interference by considering the periodic characteristic of m_k in m_i's CAN_{des}.

Definition 1: *busy sequence (BS_k) for $dhp_{GW}(i)$ message m_k*
The instance sequence that includes the maximal number of instances of m_k that can finish its transmission in m_i's CAN_{des} and arrive at m_i's CAN_{sou} in any time period of t is defined as the busy sequence of m_k.

Fig 4 describes the BS_k of m_k, which starts from T_0. T_0 indicate the arriving time of the first instance of m_k in m_i's CAN_{sou}, which equals to the finishing time of the first instance of m_k in m_i's CAN_{des}. It shows that when the first instance of m_k is finished at its WCRT and the following other instances are finished as soon as they are arrived in m_i's CAN_{des}, the number of arrived instances in m_i's CAN_{sou} in any time period of t that starts from T_0 is maximized. For BS_k, only the distance between the arriving time of the first and the second instance of m_k is equal to the closest distance between the arriving time of two continuous instances in m_i's CAN_{sou}: $(T_k - r_{s,k} + C_k)$. The distance between the arriving time of any other two continuous instances of m_k in m_i's CAN_{sou} is constrained by its period in m_i's CAN_{des}, thus it equals to T_k but not $(T_k - r_{s,k} + C_k)$. Therefore, compared with the sporadic message model proposed in [5], the definition of *busy sequence* can give a tighter bound of the interference that would be caused by m_k. Equation 1 can be used to calculate the maximal number of arrived instances for BS_k during any time period of t. For periodically arriving messages such as $shp(i)$ messages in m_i's CAN_{sou}, their busy sequences are corresponding to the periodically arriving instance sequences.

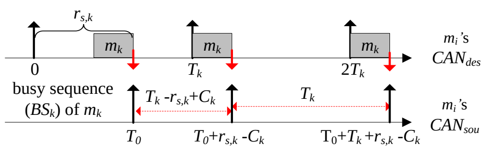


Fig 4: The busy sequence BS_k of $dhp_{GW}(i)$ message m_k

$$BS_k = \left\lceil \frac{t + r_{s,k} - C_k}{T_k} \right\rceil \quad (1)$$

However, the definition of *busy sequence* can only defines the maximal interference that would be caused by each $dhp_{GW}(i)$ message, how to define the whole interference that would be caused by all $dhp_{GW}(i)$ messages is quite another matter. The direct and intuitive assumption is that all $dhp_{GW}(i)$ messages will arrive at m_i's CAN_{sou} at the same time, thus the whole interference equals to the sum of interference that is caused by each $dhp_{GW}(i)$ message. But as Fig 5 shows, the above assumption will bring pessimism. Fig 5 describes the scheduling scenario of three messages in their CAN_{sou}, where we can find that there is distance constraint between the finishing time of different messages. The reason is that CAN messages are scheduled non-preemptively and cannot be interrupted during its transmission. Thus, for different $dhp_{GW}(i)$ messages, the arriving time in m_i's CAN_{sou} is under distance constraint. Next, we will give the Definition 2 to capture this fact.

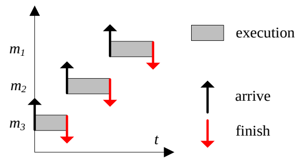


Fig 5: CAN messages' scheduling scenario

Definition 2: *the minimum distance constraint (MDC_k) for $dhp_{GW}(i)$ message m_k*

For each $dhp_{GW}(i)$ message, before its arriving time in m_i's CAN_{sou}, there will be an interval where no other $dhp_{GW}(i)$ messages can arrive, and we define this interval as the distance constraint of m_k. The theoretical low bound of the distance constraint is equal to the C_k of m_k, thus we define C_k as the minimum distance constraint MDC_k of m_k.

We only consider the MDC_k for the first instances of $dhp_{GW}(i)$ messages, but even under this assumption there is already n! different scenarios if there are n $dhp_{GW}(i)$ messages. If several instances of m_k will interfere with m_i, it becomes even more worst as the transmission of m_k will interleave with the transmission of other $dhp_{GW}(i)$ messages. Thus, it's impossible to consider the MDC_k for all instances of $dhp_{GW}(i)$ messages. In addition, MDC_k can only define the relative distance relation between m_k and any other $dhp_{GW}(i)$ messages. To get the upper bound of the whole interference that would be caused by all $dhp_{GW}(i)$ messages, we need to define the absolute distance relation for them, which means that we need to define the arriving order of all $dhp_{GW}(i)$ messages in m_i's CAN_{sou}. Therefore, we need to order all $dhp_{GW}(i)$ messages, and the objective is to find the ordered message sequence that would cause the maximal interference to m_i. To clarify the description, we use the ADC_k to indicate the absolute distance constraint between m_k and the first message of the finally ordered message sequence of all $dhp_{GW}(i)$ messages.

During the ordering process, we can only determine the ADC_k of the ordered messages, the ADC_k of the unordered messages not only depend on its own C_k, but also on the order of the already ordered messages. Therefore, the ADC_k of all $dhp_{GW}(i)$ messages can only be determined after all messages are ordered, which means that the generation all possible message sequences is a must to get the message sequence that would cause the maximal interference on m_i. The proposed ordering algorithm is based on the classical depth-first searching [9], which can generate all the possible message sequences is shown in Algorithm 1.

Algorithm 1 Ordering of All Gateway Messages

```
INPUT: Left, sum, level, Ordered
//Left: the set of unordered gateway messages, it's
initialized as the object dhp_GW(i) message set;
// sum: the number of the unordered gateway messages,
it's initialized as the number of messages in dhp_GW(i);
// level: the current level of the ordered message
sequence, it is initialized as 1;
//Ordered: the set of ordered gateway messages, it is
initialized as null;
OUTPUT: all the possible ordered message sequences
ORDERING(Left, sum, level, Ordered){
 if level < sum then
    for i=1: sum-level+1, do
       add the i-th message in Left as the last message
        of the Ordered;
        // it means to set the i-th message in Left as the
        level-th message of the ordered message
        sequence;
       Left'=Left - i-th message in Left;
       ORDERING(Left', sum, level+1, Ordered);
    end for
 else if level ==sum then
    add the only one message in Left as the last
    message of the Ordered;
    //all messages that are inserted into the Ordered in
    order represent a possible ordered message
    sequence;
    return;
 end if
 }
```

The whole execution process of Algorithm 1 will derive a searching tree, and the shape of it is shown in Fig 6 if there are n $dhp_{GW}(i)$ messages. Each path that starts from the node message of level 1 to the leaf node message of level n represents a possible ordered message sequence. Thus, the complexity of this ordering algorithm is O(n!). For each possible message sequence, ADC_k of each message can be calculated with Equation 2. It means that ADC_k of m_k equals to the sum of its own C_k and ADC_j of m_j that is located just before m_k in the finally ordered message sequence. For message in level l of the ordered message sequence, its ADC_k=0. After the determination of all ADC_k, the interference INF_k that would be caused by m_k in any time period of t that begins from the start of the ordered message sequence can be calculated with Equation 3. Compared with Equation 1, the difference is that when we calculate the times of interference that would be caused by m_k, we need to subtract ADC_k. Because in Equation 1, t indicates the start of the BS_k itself, but in Equation 3, t indicates the start of the ordered message sequence, thus we need to include the ADC_k into the calculation of the times of interference that would be caused by m_k. Please refer to Fig 8 in Section 5.1 for a concrete example. Consequently, the whole interference that would be caused by all $dhp_{GW}(i)$ messages equals to the sum of INF_k that would be caused by each $dhp_{GW}(i)$ message.

Theorem 1: *the whole interference caused by all $dhp_{GW}(i)$ messages on m_i can be upper bounded by the absolute distance correlated busy sequences of $dhp_{GW}(i)$ messages.*

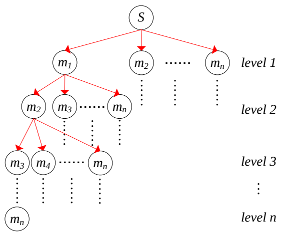


Fig 6: The searching tree for ordering of the $dlp_{GW}(i)$ messages

$$ADC_k = C_k + \underset{level(m_j)=level(m_k)-1}{ADC_j} \quad (2)$$

$$INF_k = \begin{cases} 0;\ if\ t + r_{s,k} - C_k - ADC_k \le 0 \\ \left\lceil \dfrac{t + r_{s,k} - C_k - ADC_k}{T_k} \right\rceil \cdot C_k;\ else \end{cases} \quad (3)$$

Proof: First, for each $dhp_{GW}(i)$ message, its BS_k represents the maximal number of arrived instances in m_i's CAN_{sou} in any time period, thus it upper bounds its interference on m_i. Second, for all $dhp_{GW}(i)$ messages, they cannot arrive at m_i's CAN_{sou} at the same time. The absolute distance constraint that is only considered for the first instance of all $dhp_{GW}(i)$ messages just captures this fact. But after the ordering of all $dhp_{GW}(i)$ messages, each $dhp_{GW}(i)$ message still arrives with its busy sequence, thus it can still upper bound its interference on m_i. Third, the absolute distance constraint is defined based on C_k of all $dhp_{GW}(i)$ messages, and the corresponding ordering algorithm is executed with the objective of maximizing the interference that would be caused by all $dhp_{GW}(i)$ messages. Consequently, the absolute distance correlated busy sequences of all $dhp_{GW}(i)$ messages can still upper bound their whole interference on m_i. ▪

As the complexity of the Algorithm 1 is O(n!), it cannot be used when the number of $dhp_{GW}(i)$ messages is big. Thus, we propose another simplified but pessimistic method that inspired from [10] to define the minimum distance relation between $dhp_{GW}(i)$ messages. That is we only consider the MDC_k between the first ordered $dhp_{GW}(i)$ message and all the other $dhp_{GW}(i)$ messages, and we ignore the MDC_k for all the other $dhp_{GW}(i)$ messages. As this assumption can only bring more pessimism into the interference that would be caused by all $dhp_{GW}(i)$ messages, the finally analyzed WCRT of m_i will still be safe.

5. The Proposed WCRT Analysis Approach

After the above analysis, we solved the challenge about how to define the interference for dynamic arriving gateway messages. Thus, all kinds of interference shown in Fig 3 can be analyzed now. To tackle the challenge about the inter-dependency between messages in two CAN buses, we propose the following processes for WCRT analysis of messages in gateway-interconnected CAN buses:

- First, sort all the messages inside the system in order of decreasing priority;
- Second, analyze the WCRT for each message according to the order of decreasing priority. If the current m_i is a non-gateway message, analyze its $r_{s,i}$; if the current message is a gateway message, analyze its $r_{e2e,i}$. Since when we try to analyze the WCRT of m_i, the WCRT of all the messages with priority higher than m_i in both two CAN buses are already analyzed. Therefore, all kinds of interference that would happen to m_i can be determined.

Next, we will show how to analyze the $r_{s,i}$ and $r_{e2e,i}$ in detail.

5.1 The $r_{s,i}$ analysis for non-gateway messages

The definition of *busy period* is fundamental to the WCRT analysis [7], for $r_{s,i}$ analysis of non-gateway message m_i , the level-*i* *busy period* is defined as follows：

Definition 3: *level-i busy period of* m_i

- *It starts at some time* t^s *when a message of priority i or higher is queued ready for transmission, and there are no messages of priority i or higher waiting to be transmitted that were queued strictly before time* t^s.
- *It is a contiguous interval of time during which no message of priority lower than i is able to start transmission and win arbitration;*
- *It ends at the earliest time* t^e *when the bus becomes idle, ready for the next round of transmission and arbitration, yet there are no messages of priority i or higher waiting to be transmitted that were queued strictly before time* t^e.

This time interval [t^s, t^e) is the level-*i* busy period of m_i, and $r_{s,i}$ is corresponding to the maximal level-*i* busy period $w_{s,i}$ that begins with the so called *critical instant* [11]. Inside the $w_{s,i}$ of m_i, all messages with priority higher than m_i such as $shp(i)$ or $dhp_{GW}(i)$ messages will arrive with their busy sequence pattern.

Definition 4: *the critical instant for* $r_{s,i}$ *analysis of* m_i

According to the sufficient schedulability test condition proposed in [7], we similarly define the critical instant for $r_{s,i}$ *analysis of* m_i *as follows:*

- *The arriving time of* m_i *is synchronized with all the shp(i) messages;*
- *The arriving time of* m_i *is synchronized with the ordered* $dhp_{GW}(i)$ *message set;*
- m_i *experience the maximal blocking time from* $B_{s,i}$*, where* $B_{s,i}=max(C_i , C_m , C_l)$*, m ϵ slp(i), lϵ*$dlp_{GW}(i)$*.*

Theorem 2: *when the arriving of* m_i *meets the critical instant conditions, the corresponding level-i busy period will be the maximal.*

Proof: According to the sufficient schedulability test condition proposed for one single CAN [7], when m_i experiences the maximal blocking from $B_{s,i}$, and the arriving of m_i is synchronized with all the other messages with higher priority than m_i (it represented as *shp(i)* in this paper), the level-*i* busy period will be the maximal. Based on given interference analysis in Section 3, we can easily extend this sufficient test condition to the gateway-interconnected CAN buses, where the interference caused by another type of higher priority messages that is indicated as $dhp_{GW}(i)$ also needs to be included in level-*i* busy period, and $B_{s,i}$ also needs to be extended to include the $dlp_{GW}(i)$ messages. As $dhp_{GW}(i)$ messages are asynchronous with m_i, thus when it also synchronized with the arriving of m_i, level-*i* busy period of m_i will be the maximal $w_{s,i}$. As Theorem 1 proved that the absolute distance constraint correlated busy sequences of $dhp_{GW}(i)$ messages can upper bound their interfere on m_i, thus the synchronization between the ordered $dhp_{GW}(i)$ messages and m_i means the first message of the ordered $dhp_{GW}(i)$ message sequence is synchronized with m_i. ▪

According to the definition of *critical instant*, the maximal level-*i* busy period $w_{s,i}$ can be analyzed iteratively as follows:

$$w_{s,i}^{n+1} = B_{s,i} + \sum_{j \in shp(i)} \left\lceil \frac{w_{s,i}^{n}}{T_j} \right\rceil \cdot C_j + \sum_{k \in dhp_{GW}(i)} \left\lceil \frac{w_{s,i}^{n} + r_{s,k} - C_k - ADC_k}{T_k} \right\rceil \cdot C_k \quad (4)$$

$$B_{s,i} = \max(C_i, C_m, C_l),\ m \in slp(i),\ l \in dlp_{GW}(i). \quad (5)$$

$$w_{s,i}^{0} = C_i \quad (6)$$

In Equation 4, the first part indicates the maximal blocking time, the second part indicates the interference caused by $shp(i)$ messages, and the third part indicates the interference caused by $dhp_{GW}(i)$ messages. Equation 4 will iterates until $w_{s,i}^{n+1} = w_{s,i}^{n}$.

For each possible ordered message sequence of $dhp_{GW}(i)$ messages, there will be a maximal level-*i* busy period $w_{s,i}$. Thus, $r_{s,i}$ of m_i is equal to the sum of the maximal $w_{s,i}$ corresponding with the ordered $dhp_{GW}(i)$ messages that would cause the maximal interference on m_i and C_i:

$$r_{s,i} = \max(w_{s,i}^{n}) + C_i \quad (7)$$

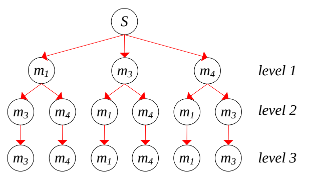


Fig 7: The searching tree for ordering of the $dlp_{GW}(6)$ messages

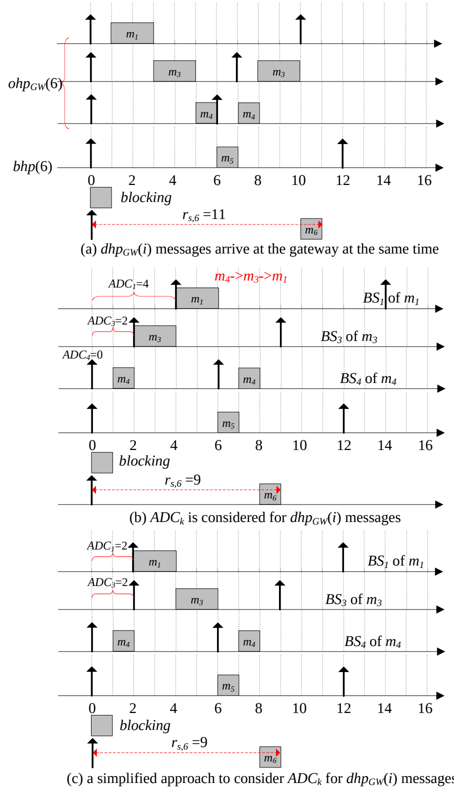


Fig 8: The ordering result for $dlp_{GW}(6)$ messages

For example, when we try to analyze the $r_{s,6}$ for m_6 in Fig 2, all kinds of messages that would contribute to the interference on m_6 are: $dhp_{GW}(6)=\{m_1, m_3, m_4\}$, $shp(6)=\{m_5\}$, $slp(6)= \{m_7\}$. Thus, to analyze $r_{s,6}$ of m_6, we need to order the three $dhp_{GW}(6)$ messages. As we already analyzed the $r_{s.i}$ of the three $dhp_{GW}(6)$ messages, therefore their busy sequences are known. The searching tree for the ordering of the $dhp_{GW}(6)$ messages is shown in Fig 7. When $dhp_{GW}(6)$ messages are ordered as m_4->m_3->m_1, they will cause the maximal interference on m_6 as shown in Fig 8(b), and m_6 will get the $r_{s,6}$=9. But under the assumption that all $dhp_{GW}(i)$ messages arrive at m_6's CAN_{sou} at the same time as shown in Fig 8(a), the $r_{s,6}$=11. As a result, by considering the MDC_k constraint for $dhp_{GW}(i)$ messages, the $r_{s,i}$ of m_i can be improved. For ordered message sequence of m_4->m_3->m_1, level and ADC_k of the three $dhp_{GW}(6)$ messages are: level(m_4)=1, ADC_4=0; level(m_3)=2, ADC_3=2; level(m_1)=3, ADC_1 =4. Fig 8(c) described the analysis scenario when the simplified but pessimistic ordering algorithm is taken to consider the MDC_k for all $dhp_{GW}(i)$ messages. In this example, when set m_4 as the first message, the $dhp_{GW}(6)$ message set will cause the maximal interference on m_6. Thus m_4 is set as the first message, and only the MDC_k between m_4 and unordered m_1 and m_3 is considered, the MDC_k between m_1 and m_3 is ignored.

5.2 The $r_{e2e,i}$ analysis for gateway messages

Considering the intractability for $r_{e2e,i}$ analysis as analyzed in Section 3, we take a division approach by ignoring the inter-dependency between interference caused by higher priority gateway messages in two CAN buses and dividing the $r_{e2e,i}$ into two separate parts as shown in Fig 9. The first part represents the WCRT of m_i in its CAN_{sou}, as it indicate the same meaning as the $r_{s,i}$ analysis of non-gateway messages, we also indicate it as $r_{s,i}$; the second part represent the WCRT of m_i in its CAN_{des} and we indicate it as $r_{d,i}$.

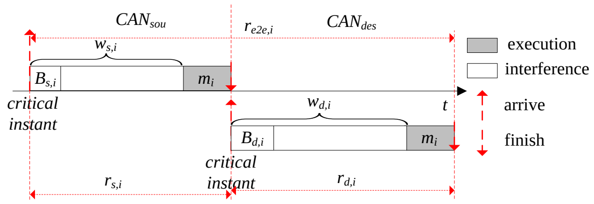


Fig 9: The division approach for $r_{e2e,i}$ analysis of gateway messages

As we ignore the gateway processing time, $r_{e2e,i}$ of m_i can be calculated as follows:

$$r_{e2e,i} = r_{s,i} + r_{d,i} \quad (8)$$

For both $r_{s,i}$ and $r_{d,i}$, it will correspond with a level-*i* busy period as shown in Fig 9. In section 5.1, we already explained how to analyze the $w_{s,i}$ and $r_{s,i}$ for m_i, next we will mostly focus on how to analyze the $w_{d,i}$ and $r_{d,i}$ for m_i. Fig 3(b) illustrates all kinds of interference that would happen to m_i when m_i is transmitted on its CAN_{des}. Compared with the analysis of $r_{s,i}$, the interference pattern of $shp_{GW}(i)$ and $dhp(i)$ messages for $r_{d,i}$ analysis is the same with that of dhp_{GW}(i) and shp(i) messages for $r_{s,i}$ analysis, respectively. The only difference is that m_i is also a gateway message that belong to the same CAN bus with $shp_{GW}(i)$ messages, thus there is also the MDC_k between m_i and all shp_{GW}(i) messages. But for $r_{s,i}$ analysis, the MDC_k only exists for $dhp_{GW}(i)$ messages. This difference add much complexity to the analysis of $r_{d,i}$, because when we try to define the interference that would be caused by all shp_{GW}(i) messages, we need to order both m_i and all shp_{GW}(i) messages. The complexity of this step is already O(n!). Furthermore, depending on the specific order of m_i, there will be several candidate positions for dhp(i) messages to start its interference. As a result, the critical instant of the maximal level-i busy period $w_{d,i}$ for the analysis of $r_{d,i}$ cannot be uniquely determined. Thus, the complexity for $r_{d,i}$ analysis of m_i is increased to O(n*n!).

Through the analysis above, we can find that compared with the analysis of $r_{s,i}$, the extra complexity for the analysis of $r_{d,i}$ comes from the fact that there is also the MDC_k between m_i and all $shp_{GW}(i)$ messages. Therefore, we take a simplified but more pessimistic approach by ignoring this point. Because under this assumption, all kinds of interference are the same for $r_{d,i}$ and $r_{s,i}$ as we described before, thus the $r_{d,i}$ analysis is transformed into the same situation with the $r_{s,i}$ analysis. Consequently, we can reuse the analysis method proposed for $r_{s,i}$. And the maximal level-i busy period $w_{d,i}$ of m_i can be calculated as follows accordingly for each possible ordered message sequence of $shp_{GW}(i)$ messages:

$$w_{d,i}^{n+1} = B_{d,i} + \sum_{j \in dhp(i)} \left\lceil \frac{w_{d,i}^n}{T_j} \right\rceil \cdot C_j + \sum_{k \in shp_{GW}(i)} \left\lceil \frac{w_{d,i}^n + r_{s,k} - C_k - ADC_k}{T_k} \right\rceil \cdot C_k \quad (9)$$

$$B_{d,i} = \max(C_i, C_m), m \in dlp(i) \quad (10)$$

$$w_{d,i}^0 = C_i \quad (11)$$

In Equation 9, the first part indicates the maximal blocking time, the second part indicates the interference caused by $dhp(i)$ messages, and the third part indicates the interference caused by $shp_{GW}(i)$ messages. Equation 9 will iterates until $w_{d,i}^{n+1} = w_{d,i}^n$.

And $r_{d,i}$ of m_i can be calculated as follows:

$$r_{d,i} = \max(w_{d,i}^n) + C_i \quad (12)$$

6. Experiment

In Section 5, we already proved that the proposed method is correct and can get the safe upper bound of WCRT for messages in gateway-interconnected CAN buses. Next, we will further show the usability of the proposed method by comparing it with a full space searching based simulator. The small message set shown in Fig 2 is used as the experimental message set, and parameters of the messages are shown in Table 1. The full space searching based simulator is implemented by searching all the possible execution scenarios, where both the message combination and the offset relation between messages are considered. Thus, its complexity grows exponentially. That's why we can only use a small message set to verify the usability of the proposed message set. In this experiment, both the Algorithm 1 and the simplified ordering algorithm are used to order the gateway messages.

Table 1: Parameters of the experimental messages set

Message's Affiliation				P_i	T_i	C_i
CAN_1	ECU_1	m_1	GW	1	12	2
		m_2	NGW	2	12	1
	ECU_2	m_3	GW	3	12	2
		m_4	GW	4	12	1
		m_8	NGW	8	12	1
CAN_2	ECU_3	m_5	GW	5	12	1
	ECU_4	m_6	GW	6	12	1
		m_7	NGW	7	12	1

Table 2: Experimental results

Message's Property		Simulator's Result		Analysis Result-*Alg 1*		Analysis Result-*simplified*	
		$r_{s,i}$	$r_{e2e,i}$	$r_{s,i}$	$r_{e2e,i}$	$r_{s,i}$	$r_{e2e,i}$
m_1	GW	4	**7**	4	**8**	4	**8**
m_2	NGW	**5**	-	**5**	N	**5**	N
m_3	GW	6	**9**	7	**13**	7	**13**
m_4	GW	7	**10**	7	**13**	7	**13**
m_5	GW	7	**9**	7	**15**	7	**15**
m_6	GW	7	**9**	9	**19**	9	**19**
m_7	NGW	**8**	-	**12**	-	**12**	-
m_8	NGW	**9**	-	**12**	-	**12**	-

Table 2 shows the analysis result of the proposed methods and the simulator's result. First, the WCRT analysis is the same for all messages when the ordering of gateway messages is implemented with Algorithm 1 or the simplified ordering algorithm, which means that the simplified ordering algorithm is also useful in improving the WCRT of messages. Second, we can find that for $r_{s,i}$ analysis, the analysis result is close to the simulator's result, but for $r_{e2e,i}$ analysis, the pessimism is relative bigger. The main reason is that we took a division approach to analyze the $r_{e2e,i}$ and ignored the inter-dependency between interference caused by higher priority gateway messages in two CAN buses.

For this message set, the simulator took about 4 minutes to get the final result, and the other two analysis methods took less that 1 second to get the final result. But in another experiment, we assigned the same 8 messages into 6 ECUs (we assigned m_3 and m_4 in ECU_1 into the other two ECUs separately), the simulator ran for 3 days but still cannot get the final result. But for the other two analysis method, the execution time is the same. The reason is that the messages assigned into the same ECU are synchronized, but for messages that are assigned in different ECUs, there will be offset relations among them, thus the execution scenario that needs to be checked by the simulator will be increased exponentially.

7. Conclusions and Future Work

The main contribution of this paper is a WCRT analysis method that tackled the following three challenges and can get the safe upper bound of WCRT for messages in gateway-interconnected CAN buses:

- the dynamic arriving gateway messages;

- the inter-dependency between messages in two CAN buses that is brought by gateway messages;
- the inter-dependency between interference caused by higher priority gateway messages in two CAN buses;

The correctness of the proposed method is proved, and with a small message set, its usability is also verified by comparing it with a full space searching based simulator.

According to the line of research for one single CAN, the offset assignment can improve the WCRT of messages to a large extent. Therefore in the future, we will apply the offset assignment for messages in gateway-interconnected CAN buses to improve their WCRT.

Reference

1) Navet, N., Simonot-Lion, F.: A review of embedded automotive protocols, in Automotive Embedded Systems Handbook(N. Navet and F. Simonot-Lion, Eds.), CRC Press /Taylor and Francis(2008).

2) Sangiovanni-Vincentelli, A., Di Natale, M.: Embedded system design for automotive applications, *IEEE Journal of Computer*, Vol.40, No.10, pp.42-51 (2007).

3) Renesas. http://am.renesas.com/applications/automotive/ solution/networking/.

4) Sommer, J. and Andblind, R.: Optimized resource dimensioning in an embedded CAN-CAN gateway, *Proc. Int. Sym. on Industrial Embedded Systems*, pp.55–62(2007).

5) Davis, R. and Navet, N.: Traffic shaping to reduce jitter in Controller Area Network (CAN), *Proc. Work-in-Progress Session of ECRTS 2012*. Available at: http://www-users.cs.york. ac.uk/~robdavis/papers/TrafficShapingCANv2.1.pdf

6) Tindell, K.W., Hansson, H. A. and Wellings, J.: Analysing real-time communications: Controller Area Network (CAN), *Proc. Real-Time Systems Symposium*, pp.259-263(1994).

7) Davis, R.I., Burns, A., Bril, R.J. and Lukkien, J.J.: Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised, *Real-Time Systems*, Vol.35, No.3, pp.239–272 (2007).

8) Sojka, M., Písa, P., Spinka, O., Hanzalek, Z.: Measurement automation and result processing in timing analysis of a Linux-based CAN-to-CAN gateway, *Proc. IEEE Int Conf on Intelligent Data Acquisition and Advanced Computing Systems*, pp.963-968(2011).

9) Cormen, T.H., Leiserson, Ch.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill (2001). Section 22.3: Depth-first search, pp. 540–549.

10) Rox, J., Ernst, R.: Exploiting inter-event stream correlations between output event streams of non-preemptively scheduled tasks, *Proc. Design, Automation&Test in Eurepean Conference&Exhibition*, pp.226-231(2010).

11) Liu, C.L., Layland, J.W. Scheduling algorithms for multiprogramming in a hard-real-time environment, *Journal of the ACM*, Vol.20, No.1, pp.46-61(1973).

Acknowledgments This work was supported by the National Science Foundation of China (Grant No. 61173036).