Schedulability Analysis of Ethernet AVB Switches

Unmesh D. Bordoloi, Amir Aminifar, Petru Eles, Zebo Peng
Department of Computer and Information Science, Linköping University, Sweden
{unmesh.bordoloi, amir.aminifar, petru.eles, zebo.peng}@liu.se

***Abstract*— Ethernet AVB is being actively considered by the automotive industry as a candidate for in-vehicle communication backbone. However, several questions pertaining to schedulability of hard real-time messages transmitted via such a switch remain unanswered. In this paper, we attempt to fill this void. We derive equations to perform worst-case response time analysis on Ethernet AVB switches by considering its credit-based shaping algorithm. Also, we propose several approaches to reduce the pessimism in the analysis to provide tighter bounds.**

I. Introduction

With a proliferation of applications and the sheer volume of data that is expected to be transmitted in automotive networks as a result, traditional fieldbuses like FlexRay/CAN will not be able to offer the required bandwidth. As response to this challenge, Ethernet is being considered by the industry as an alternative bus vehicular protocol. In particular, the suitability of the IEEE 802.1 Audio/Video Bridging (AVB) standard is being actively discussed [1]. This is because (i) of its status as an official standard and (ii) it is already widely used in several other industrial segments. Ethernet AVB is a switch-based protocol and the switches rely on a credit-based shaping algorithm to regulate traffic flow. Several messages may share the same priority and such messages are said to belong to a class in Ethernet AVB (see Section II). However, its applicability to hard real-time applications hinges on whether or not tight worst-case response times (WCRT) of messages transmitted via an AVB Ethernet switch can be computed efficiently.

Our Contributions: As opposed to typical scheduling policies and bus protocols like CAN [2], Ethernet AVB is *not* a work-conserving (non-idling) policy due to the traffic shaper mechanism. Ethernet AVB using a non-preemptive scheme, a message may experience several points of blocking by the low priority messages (see Figure 3(a) for an example). This is unlike non-idling protocols, e.g., CAN bus, where there might be at most one such blocking.

The busy period analysis typically applies to non-idling scheduling policies, and therefore, it is enough to consider one blocking by the lower priority messages. However, the previous work [3] defines the busy period concept for the Ethernet AVB (which is not a work-conserving protocol) and proposes to use this definition to perform worst-case response time analysis, without providing theoretical support regarding the fact that this scenario actually leads to the worst-case response time. Moreover, the authors assume, without demonstrating, that considering one blocking is sufficient. Therefore, the correctness of the previous work is not evident.

In this paper, we formally address several challenges that are specific to Ethernet AVB analysis, none of which was discussed previously: **(i)** The number of blocking by the lower priority messages is discussed. We provide a formal proof, of course under our set of assumptions, that once we account for the blocking by the traffic shaper, it is only then safe to consider one blocking from the lower priority messages to compute the worst-case response time (Section IV). **(ii)** The state of the traffic shaper is important in computing the WCRT. We discuss that under a new seemingly pessimistic definition of the WCRT (Section V) that assumes that the credit recovery after the transmission of the last message be included in the WCRT, the state of the traffic shaper at the start of the busy period may be ignored. **(iii)** Discarding the impact of the traffic shaper of the higher priority classes is safe while computing the WCRT of a message. We discuss that the shaper may only postpone the transmission of a message, which in turn may lead to less interference by the higher priority messages for some instances of the message under analysis (Section VI). However, the effect of the traffic shapers will be taken into account in order to improve the analysis. **(iv)** Ignoring the FIFO queues for the higher priority classes is safe when we discard the impact of their traffic shaper. This is due to the fact that the WCRT depends on the amount of interference rather than the order of messages, where no shaping algorithm is considered (Section VI).

Beyond this, our proposed worst-case response time analysis accounts for the impact of traffic shaper on higher priority messages as well as the overlapping between higher priority messages and idle times on the bus, which considerably reduces the pessimism involved.

Related Work: Of late, there has been a growing interest in *formal* timing and scheduling analysis of messages communicating over Ethernet AVB switches. Above, we have already discussed one of the previous work (i.e., [3]). It should be mentioned that there are other approaches [4] in the literature as well but they do not consider the interference from higher priority messages in a formal way. This renders them inapplicable for WCRT analysis.

Even earlier approaches, like [5], [6] and [7], restrict the computation of timing results on a per-class basis without distinguishing between different messages in the class. In the context of automotive applications, it is typical to have over thousands of signals [8] communicating over the fieldbus. It is inevitable that multiple message streams will be mapped to the same class. Unfortunately, with most

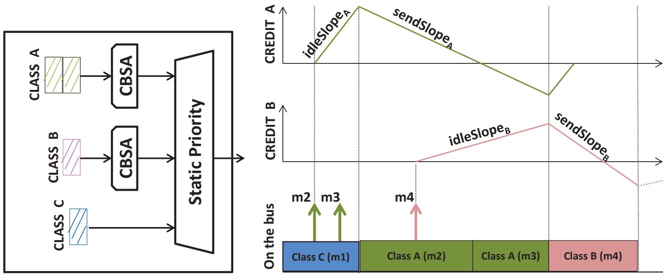

Fig. 1. Snapshot of an exemplery AVB switch.

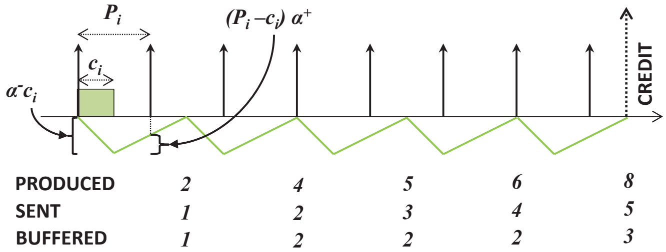

Fig. 2. Violating the necessary condition implies an infinite buffer eventually.

existing bodies of work [5], [6], [7], it is not possible to bound the latencies of individual message streams.

It is noteworthy that several variations of Ethernet AVB are being proposed in the community as we write this paper, although not yet accepted as part of the standard. An example would be [9], where the authors propose to introduce a higher priority class (which does not undergo credit-based shaping, but works according to priority) on top of the existing traffic-shaper classes for time-sensitive traffic flows. The techniques in this paper may be generalized to such a variant. In fact, most likely they remain valid for the existing traffic-shaper classes with minor adjustments.

Furthermore, analysis methods have also been proposed for other Ethernet variants like the weighted round robin scheduling [10] for Ethernet and the AFDX Ethernet protocol [11] among others. However, WCRT analysis for these variants are very different from Ethernet AVB and a more elaborate discussion is out of scope of this paper.

II. System Model

Our system model consists of two major components — (i) the Ethernet AVB specifications and (ii) characteristics of frames (or messages).

Ethernet AVB: Ethernet AVB allows strict priority non-preemptive scheduling for messages. All messages must be assigned a priority and more than one message may have the same priority. The set of messages assigned the same priority are said to belong to the same traffic class. Messages within each traffic class follow FIFO order.

On top of the priorities, Ethernet AVB adds a credit-based shaping algorithm (CBSA) for at least two traffic classes. These traffic classes are typically denoted as “A” and “B”, with A of higher priority than B. Thus, a credit level is associated with each of these traffic classes. For simplicity of elucidation, in this paper we will consider that only two classes are shaped by a traffic shaper. It is important to note, however, that all our results can be generalized to an arbitrary number of classes with traffic shaper, except for the improvement I in Section VI-B, which is specific for class B.

Ethernet AVB CBSA stipulates the following. Frames in class A and B may be transferred only if the corresponding credit level is zero or higher. The credit levels of traffic class A (and traffic class B) are replenished at a constant rate, henceforth denoted by α_A^+ (and respectively, α_B^+). This slope is called the *idleSlope* of the corresponding class. It is important to note that the credits are replenished only when (i) the messages of the corresponding traffic class are waiting for the transmission or (ii) no more frames of the corresponding traffic class are waiting but the credit is negative. On the other hand, if the credit level is positive and no more frames of the corresponding traffic class are waiting, the credit is immediately reset to zero. Finally, note that when a message from class A (or class B) is transmitted the credits of the corresponding class are decremented at the rate α_A^- (and respectively, α_B^-). This rate is called the *sendSlope*.

Figure 1 shows an example of an Ethernet AVB switch with 3 priorities. Class A messages (m_2 and m_3) have highest priority. Class B messages (m_4) has lower priority than Class A messages. Class C is not shaped and its messages (m_1) have the lowest priority. More specifically, in the figure, we show a snapshot where two messages m_2 and m_3 of class A are in the buffer for a certain length of time before being transmitted on the bus. Message m_1 from a lower priority class (class C) blocks m_2 and m_3 immediately when they arrive since it is already on the bus. This is because Ethernet AVB messages are transmitted non-preemptively. As soon as m_2 is blocked, credit A starts accumulating. Thereafter, m_2 and m_3 are transmitted after m_1 completes transmission. During this time credit A is decremented. Messages within each traffic class follow FIFO order and hence, m_2 is transmitted before m_3 in our example. Similarly, credit in class B starts to accumulate as soon as message m_4 is blocked and the message is eventually transmitted.

Messages: We consider a set of frames $\mathbf{H}$ where each frame $m_i \in \mathbf{H}$ is characterized by the following parameters.

- **Period**: The rational period P_i, denotes the time interval after which a new instance of m_i is produced.
- **Deadline**: The deadline D_i, of a frame m_i is the relative time since the production of m_i until the time by which the transmission of m_i must end.
- **Priority:** The priority of each message m_i, that is used as one of the mechanisms to resolve bus access contentions, is assumed to be known. The set of messages with higher, same, and lower priority compared to m_i are denoted by $hp(m_i)$, $sp(m_i)$, and $lp(m_i)$, respectively.
- **Transfer time**: Based on the size of each frame and the bandwidth of the bus, we are given the transfer (communication) time C_i of the frame m_i.

III. Bounds on Workload

In this section, we derive bounds on allowed load of messages that may be transmitted over an Ethernet AVB switch, which might be of interest for practitioners/engineers during early stages of the design cycle, as *necessary* conditions, on account of their simplicity. First, we will present the condition for schedulability of a single message and then extend the result for all messages in a class. This will be followed by two key insights, that will be used later in the paper.

A. For single message

We state our result in the form of the Lemma below.

Lemma 1. *The necessary condition for schedulability of message m_i is*

$$\frac{C_i}{P_i} \le \frac{\alpha^+}{\alpha^+ + \alpha^-}. \tag{1}$$

Proof. The correctness of this bound may be proven by contradiction as follows. Let us consider that message m_i is the only message in a system and it arrives with a period P_i. As soon as the message arrives, it is transmitted and amount of credit $C_i\alpha^-$ is consumed. We will show that even in this simple setup, Equation 1 is a necessary condition and this will imply that this condition holds true in the general case as well. Contradicting our claim, let us now say that Equation 1 is not satisfied. Considering the complement of Equation 1 and rearranging the terms gives us the following inequality,

$$C_i\alpha^- > (P_i - C_i)\alpha^+. \tag{2}$$

The term on the left quantifies the credit that is lost during transmission. The term on the right quantifies the credit gained after the transmission but before the next instance arrives. Equation 2, thus, means that the credit cannot recover to zero before the next instance of the message arrives. This is illustrated in Figure 2. Let us say that the second instance has to wait ϵ units of time for the credit to recover.

For the next instance, this waiting of ϵ units of time is doubled and eventually it will become infinite and then, the message becomes unschedulable. Our example visualizes this by showing the size of buffer (at each time instant when the credit recovers to zero). Intuitively, for a system to be schedulable, this necessary condition is based on the fact that the bandwidth provided by the traffic shaper should be more than or equal to the demand by the message. □

B. For a class

We extend the above result on a per class basis. Let us denote a class with X, i.e., class X represents either class A or class B. The result for the class may be stated as follows.

Theorem 1. *The necessary condition on a per class basis is as follows,*

$$\sum_{m_i \in class\ X} \frac{C_i}{P_i} \le \left(\frac{\alpha_X^+}{\alpha_X^+ + \alpha_X^-}\right). \tag{3}$$

Proof. For the proof, let us consider a time interval equal to the hyper-period $\Pi = \operatorname*{lcm}_{m_i \in \text{class } X} (P_i)$. The total workload generated in Π by messages in class X is given by

$$\sum_{m_i \in \text{class } X} C_i \frac{\Pi}{P_i}. \tag{4}$$

This workload in the hyper-period consumes a total credit equivalent to $\sum_{m_i \in \text{class } X} C_i \frac{\Pi}{P_i} (\alpha_X^-)$ and thus, needs the following time to recover from the consumed credit back to a credit level of zero,

$$\sum_{m_i \in \text{class } X} C_i \frac{\Pi}{P_i} \left(\frac{\alpha_X^-}{\alpha_X^+}\right). \tag{5}$$

Taking the sum of the two terms above, we have the total time within hyper-period Π where the resource at class A is either busy transmitting the workload or blocked by the traffic shaper. This sum should be less than the hyper-period otherwise there will be a non-zero finite delay into the next hyper-period before transmission can begin in the second hyper-period. Analogous to the proof for Lemma 1, where we discussed only one message, this shift will eventually lead to an infinite backlog and response time. This condition is shown below,

$$\sum_{m_i \in \text{class } X} C_i \frac{\Pi}{P_i} \left(1 + \frac{\alpha_X^-}{\alpha_X^+}\right) \le \Pi. \tag{6}$$

Rewriting this equation, we obtain Equation 3. □

From Equation 3, it may be concluded that *the utilization of a class is bounded above by* $\left(\frac{\alpha_X^+}{\alpha_X^+ + \alpha_X^-}\right)$.

C. Bounds on the workload

Above, we discussed the necessary conditions for schedulability based on the workload. As a corollary, we now present two key insights regarding the workload in a specially defined interval of time that begins and ends with zero credits. As will become apparent later in the paper, analyzing the *transmitted* workload in such an interval is a crucial component to develop the worst-case response time (WCRT) analysis.

Corollary 1. *Considering a class X, for an interval of time t that starts with zero credits and ends with zero credits, if the queue for that class remains non-empty during this interval, the workload transmitted in the time interval t is exactly equal to* $x = t \cdot \left(\frac{\alpha_X^+}{\alpha_X^+ + \alpha_X^-}\right)$.

Proof. Let us denote the workload that is transmitted as x. During the time interval t, the credit lost is then given by $x \cdot \alpha_X^-$ while the credit gained is given by $(t - x) \cdot \alpha_X^-$. The summation of these two terms should be zero, i.e.,

$$-x \cdot \alpha_X^- + (t - x) \cdot \alpha_X^+ = 0$$

Re-arranging the terms of the above equation, we will find that $x = t \cdot \left(\frac{\alpha_X^+}{\alpha_X^+ + \alpha_X^-}\right)$ and this proves the claim of the corollary. □

Corollary 2. *Considering a class X, for an interval of time t that starts with zero credits and ends with zero credits, the workload transmitted in the time interval t is less than or equal to* $t \cdot \left(\frac{\alpha_X^+}{\alpha_X^+ + \alpha_X^-}\right)$.

Proof. Note that if the queue is empty and the credit is positive, the credit is reset to zero, while it is not reset to zero if the credit is negative. Let us denote the workload that is transmitted in an interval of length t by x. Since during $(t-x)$ the queue might become empty, the credit recovered, denoted by $y \cdot \alpha_X^+$ with $y \leq (t-x)$, is more than[1] or equal to the credit exhausted during transmission time x (i.e., $x \cdot \alpha_X^-$),

$$x \cdot \alpha_X^- \leq y \cdot \alpha_X^+ \overset{y \leq (t-x)}{\Longrightarrow} x \cdot \alpha_X^- \leq (t-x) \cdot \alpha_X^+.$$

Re-arranging the terms of the above equation, we find,

$$x \leq t \cdot \left(\frac{\alpha_X^+}{\alpha_X^+ + \alpha_X^-}\right).$$

□

IV. Factors leading to WCRT

The worst-case response-time (WCRT) analysis for messages in class A and class B of Ethernet AVB switches raises different challenges, when compared to other fixed priority protocols like CAN [12]. This is because of the following factors.

First and foremost, the messages have a traffic shaper that can block messages from transmission even if they are ready to be transmitted and the bus is idle. In contrast, CAN is a non-idling bus and there is no traffic-shaper in CAN that needs to be considered. Second, in the CAN protocol, at most one lower priority message may block a higher priority message. However, in AVB Ethernet, as we will discuss in Section IV-C, due to the interaction with the traffic shaper, it is possible that several lower priority messages are transmitted while a higher priority message is waiting. Third, there are several messages sharing the same priority (messages that belong to the same class) and these messages are transmitted in a FIFO fashion. Finally, messages in class B are also delayed due to interference by higher priority messages in class A. However, these messages are shaped by their own traffic shaper. Thus, the factors influencing the WCRT include (i) the maximum possible interference from the traffic shaper, (ii) maximum interference due to FIFO scheduling with other messages in the same class (iii) blocking by lower priority messages due to its non-preemptive nature and (iv) interference due to higher priority messages.

In this section, we will describe in detail the first three factors above and how they influence the WCRT. The last factor concerns only class B and this will be discussed in Section VI. However, before these details, we must clarify some assumptions.

Note that, similar to the CAN bus analysis [12], a busy period analysis is required for the Ethernet AVB messages. For details of the busy period analysis, we refer the reader to an excellent description in [12]. It suffices here to state that it is not enough to analyze only one instance of a message and, instead, all instances of a message that arrive during the *busy period* must be analysed. However, for the highest priority class, we will show (Section V) that it is safe to consider only the response time for one instance. Section VI extends the analysis to consider several instances with the busy period based analysis.

In this work, we assume that $D_i \leq P_i$. This assumption is reasonable because in this paper, we are concerned with the delay on one Ethernet switch and not with the end-to-end delay over the system. Extending our work to compute the end-to-end delay over multiple hops may be done in line with known holistic analysis tools [13], [14], but it is out of scope of this paper.

A. Initial credit for the traffic shaper

As discussed, a message in class A or B may be transmitted if and only if the corresponding credit is either zero or positive. To compute the WCRT, we have to consider that the value of the traffic shaper credit is minimal at the critical instant. With the critical instant we refer to the time instant where, if a message is released, it will suffer from the worst-case response time. In the following, we provide a lower bound LB_{credit} for the worst-case negative credit for a class X, where X is either class A or class B.

$$LB_{credit} = \alpha_X^- \cdot \max_{m_i \in \text{class } X} \{C_i\} \tag{7}$$

The result follows from the definition of the AVB Ethernet protocol that states that a message can only be transmitted when the credit is not negative. Thus, a second message m_j in any class may not be transmitted when the credit is negative due to prior transmission by another message m_i in the same class. This implies that the credit may not decrease any further than the maximum negative credit, $-\alpha_X^- \cdot C_i$, by an individual message m_i. Considering the worst-case amongst all messages gives us the above result.

B. Impact of FIFO policy

Under the assumption of deadline less than or equal to the period, the maximum interference a message can experience in a schedulable system is equal to the sum of the transmission times of all messages in the same FIFO queue [15]. This is because of the fact that in a schedulable system with deadline less than or equal to period, only one instance of a message can block the message under analysis. Observe that in such systems an instance of a message finishes its transmission before the next instance is released. It should be noted that this scenario occurs when all messages in the same FIFO queue arrive just before the message under analysis.

[1]The (positive) credit is reset to zero, when the queue becomes empty.

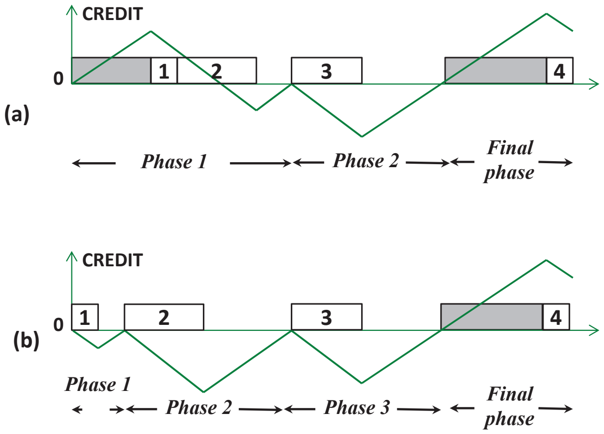

Fig. 3. To analyze worst-case interference from lower priority message, we look at the blockings in terms of phases. Figure(a) shows a pattern starting with lower priority message. Figure(b) shows a pattern starting without lower priority message.

C. Blocking by lower priority messages: Class A

As mentioned before, the analysis of the AVB Ethernet protocol needs to consider another additional factor compared to a protocol like CAN. This is because the bus may be idle due to the traffic shaper and a higher priority message *potentially* may experience several blockings from the low priority messages during these idle intervals.

However, we will show that it is sufficient to consider the blocking time by only *one* lower priority message if we include the blocking time due to the traffic shaper. In other words, the blocking time due to the traffic shaper of the corresponding class subsumes the additional blockings and hence, interference from only one of the blockings from the lower priority messages needs to be considered. For clarity of explanation, let us first consider only class A messages implying that we need not factor in interferences from higher priority messages. The next section extends the proof to class B.

Towards the proof, observe that for any transmission, starting with zero credit[2], the traffic shaper credit accumulation follows a pattern that consists of a sequence of phases. A phase is either (i) a time interval that begins with zero credit and ends with zero credit or (ii) a time interval that begins with zero credit and ends with the transmission of the message under analysis. For illustration of phases, let us look at Figure 3(a), where we show three phases. In the figure, the blockings by low priority messages are depicted by shaded boxes, while the messages from the same class are shown in white. The intuition behind our proof is that in all phases the traffic shaper blocking subsumes the blocking by lower priority message except the final phase. Hence, considering only one blocking is enough.

Non-Final Phases: Each phase, except the final phase, has three components. Note that any of the components may be of length zero. The first component of a phase is the time interval that starts with the blocking by, at most, one low priority message. In phase 1 of Figure 3(a), this component is non-zero and in phase 1 of Figure 3(b) this is zero because it represents a scenario where there was no blocking by a lower priority message. For phase 2 in Figure 3(a) and for phase 2 and phase 3 in Figure 3(b), we can observe that the first component is zero. The first component, if present, is followed by the second component that marks the transmission of messages in the same class during which the credit decreases. In phase 1 of Figure 3(a), this component involves the time for transmission of two messages after the transmission of the low priority message. Phase 1 of Figure 3(b) starts with this component marking the transmission of one message and, again, the credit decreases during this component. The third component consists of credit recovery to zero.

We will now show that the length of each of these phases, except the final one, is independent of the blocking time by lower priority messages. We show this for phase 1 but the same result holds for the rest. The length of a phase, denoted by l, may be given by the following summation of the three components that we discussed above.

$$l = C_{\text{lp}} + C_{\text{sp}} + x \tag{8}$$

C_{lp} is transmission time of the lower priority message (first component); C_{sp} is the sum of transmission time of the messages with same priority (second component) and x is the credit recovery time (third component).

As the credit is zero at the end and the beginning of the phase, the total gain and loss in credit since the beginning of the phase must sum up to zero,

$$\begin{aligned} 0 &= C_{\text{lp}} \cdot \alpha^+ - C_{\text{sp}} \cdot \alpha^- + x \cdot \alpha^+, \\ x &= C_{\text{sp}} \cdot \frac{\alpha^-}{\alpha^+} - C_{\text{lp}}. \end{aligned} \tag{9}$$

Substituting this value of x in Equation 8, we have $l = C_{\text{sp}} \cdot \left(1 + \frac{\alpha_A^-}{\alpha_A^+}\right)$. This equation for the length of a phase shows that for all phases, except the last one, the blocking by the lower priority messages need not be explicitly added.

Final Phase: The final phase, however, is different from others since the credit does not necessarily need to be replenished and then recovered to zero. This is because the phase ends as soon as the message under analysis has completed its transmission. In the scenario for the worst-case response time, the final phase should start with the longest low priority message blocking the message under analysis followed by the message under analysis.

Note that to construct the worst-case scenario, we must consider that no other message from the same class is transmitted in the final phase. This can be proved by contradiction. Let us consider a scenario where we have, in the final phase, several messages which belong to the same class before the message under analysis. Let us also assume that this is the worst-case scenario.

Note that since this is the final phase, before the message under analysis starts transmitting, the credit should be positive (otherwise it cannot be the final phase). If we

[2]The reason for starting with zero credit will become apparent as we discuss a "Refinement" in our definition of WCRT in the next Section.

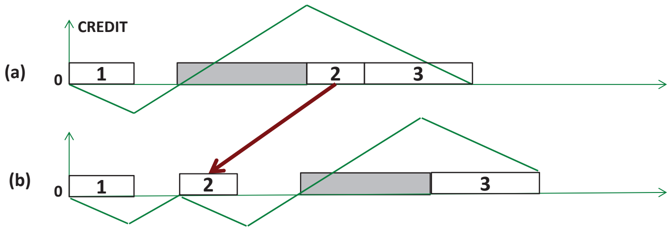

Fig. 4. In the worst-case scenario, the final phase contains only one message.

consider another scenario by moving these messages and creating several new phases right before the final phase, we have a worse scenario than the initial one, i.e., a scenario in which the response time of the message under analysis is larger. This can be verified by the fact that, for each message m_i in the final phase, creating a new phase leads to $C_i \cdot \left(\frac{\alpha^-}{\alpha^+}\right)$ extra interference. An example is provided in Figure 4. The message under analysis is m_3 and m_1, m_2 and m_3 are messages of the same class. The scenario in Figure 4(a) shows a final phase consisting of two messages m_2 and m_3 of the same class. Moving message m_2 to a new phase before the final phase leads to a worse delay for message m_3 as shown in Figure 4(b). This contradicts the assumption that the initial scenario represents the worst-case.

With this, we have now shown that, in the case of class A, for the worst-case scenario, the final phase is one where only one message (the message under analysis) is transmitted apart from the lower-priority message. We have also shown that for all previous phases blocking needs not be formally added. Together, this concludes the claim that the blocking from the lower-priority message needs to be considered only once.

D. Blocking by lower priority messages: Class B

We now show that even for class B messages, where higher priorities interfere, it is enough to consider only one blocking by a low priority message. As this proof bears many similarities to the one discussed above for class A, we provide a succinct proof sketch here.

Once again, our goal is to show that the length of any phase, except only the last phase, is independent of the lower priority message. The length of any phase l is now given by the following,

$$l = C_{\text{lp}} + C_{\text{sp}} + C_{\text{hp}} + x, \tag{10}$$

where C_{lp} is the blocking by the lower priority messages, C_{sp} is the blocking by the same priority messages, C_{hp} is the blocking by the higher priority message, and x is the time taken for credit recovery — all terms are in reference to the phase l. Again, we know that during such a phase the total credit gain is zero. Similar to class A, hence, we have the following.

$$\begin{aligned} 0 &= (C_{\text{lp}} + C_{\text{hp}} + x) \cdot \alpha_B^+ - C_{\text{sp}} \cdot \alpha_B^-, \\ C_{\text{sp}} \cdot \left(\frac{\alpha_B^-}{\alpha_B^+}\right) &= C_{\text{lp}} + C_{\text{hp}} + x \end{aligned} \tag{11}$$

Rewriting Equation 10, we can now show that the length of a phase l is independent of the lower priority messages,

$$l = C_{\text{sp}} \cdot \left(1 + \frac{\alpha_B^-}{\alpha_B^+}\right). \tag{12}$$

This result states that the low priority messages may at most interfere with the transmission of a higher priority message once (i.e., at the beginning of the final phase).

V. WCRT Analysis for Class A

The credit of the traffic shaper is at the minimum when the message under analysis arrives. As discussed in Section IV-A, the minimum credit is bounded. The worst-case credit LB_{credit} (see Equation 7) for class A, may be recovered back to zero in $\frac{LB_{credit}}{\alpha_A^+}$ time units. Expanding LB_{credit}, the time needed for the credit to recovery to zero is obtained,

$$I_i^{initialcredit} = \max_{m_j \in \text{class } A} \{C_j\} \left(\frac{\alpha_A^-}{\alpha_A^+}\right), \tag{13}$$

where the largest low priority message arrives just before the message under analysis and blocks it.

Second, as it is shown in Section IV-C, it is sufficient to consider only one blocking in the worst-case scenario and the interferences by the rest are subsumed by the traffic shaper blocking. The maximum blocking thus is,

$$I_i^{blocking} = \max_{m_j \in \text{lp (class A)}} \{C_j\} \tag{14}$$

Finally, all other messages in the same class arrive just before the message under analysis in the critical instant, assuming a FIFO behavior. As mentioned in Section IV-B, in a schedulable system, the message under analysis may not be delayed by more than one instance of each message and, therefore, it is enough to consider all the messages in the same class arrive just before the message under analysis. Furthermore, the blocking by the traffic shaper that arises due to messages in the same class needs to be considered. The amount of interference from the messages in the same class assuming FIFO and the traffic shaper is given in the following,

$$\sum_{m_j \in \text{class } A} C_j \left(1 + \frac{\alpha_A^-}{\alpha_A^+}\right). \tag{15}$$

As discussed, the credit does not need to be recovered after the transmission of the message under analysis m_i in the final phase. Therefore, the above equation can be modified as follows,

$$I_i^{classA} = \sum_{m_j \in \text{class } A} C_j \left(1 + \frac{\alpha_A^-}{\alpha_A^+}\right) - C_i \left(\frac{\alpha_A^-}{\alpha_A^+}\right). \tag{16}$$

Following the above discussion, we are now in position to list the terms that must be summed up to compute the WCRT for a message in class A,

$$R_i^{\text{w}} = I_i^{initialcredit} + I_i^{blocking} + I_i^{classA}. \tag{17}$$

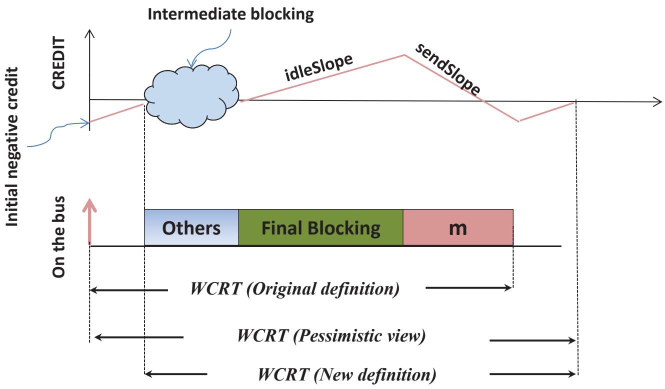

Fig. 5. Previous and new response-time definitions.

To sum up,

$$R_i^{\mathrm{w}} = \max_{m_j \in \text{class } A} \{C_j\} \left(\frac{\alpha_A^-}{\alpha_A^+}\right) + \max_{m_j \in \text{lp (class A)}} \{C_j\} + \sum_{m_j \in \text{class } A} C_j \left(1 + \frac{\alpha_A^-}{\alpha_A^+}\right) - C_i \left(\frac{\alpha_A^-}{\alpha_A^+}\right). \tag{18}$$

Refinement: We will now reduce the pessimism in the above result by redefining the worst-case response time to include the traffic shaper blocking for the message under analysis in the final phase, i.e., we take a pessimistic view (see Figure 5) and require credit recovery for the last message in the final phase. This seems counter-intuitive at first because we are artificially stretching the end point of the message transmission.

However, it is important to note that with this definition of the response time, in a schedulable system, the credit at the critical instant (start of the busy period) cannot be negative. If it were the case, this means that a message has violated its deadline under the new definition (since we consider constrained deadlines) and the system is not schedulable (that contradicts our assumptions). Hence, for any schedulable system, with our new definition (see Figure 5) of the response time, we can safely ignore the term for the negative credit at the critical instant and Equation 18 can be simplified,

$$R_i^{\mathrm{w}} = \max_{m_j \in \text{lp (class A)}} \{C_j\} + \sum_{m_j \in \text{class } A} C_j \left(1 + \frac{\alpha_A^-}{\alpha_A^+}\right). \tag{19}$$

Observe that the worst-case response-time given by Equation 19 cannot be larger than the worst-case response-time calculated by Equation 18. Further, from Equation 19 one may observe that the worst-case response times of all messages in class A are the same.

VI. WCRT Analysis for Class B

In this section, we will discuss the response time analysis for messages from class B. Unlike class A, messages from class B are blocked also by messages from the higher priority class. Moreover, performing the analysis only for one instance of a message from class B is not sufficient. Rather, the response time for instances of messages that are produced within the *busy period* [16] must be analyzed. The maximum among them gives us the WCRT. We proceed to compute the busy period and the response times.

Let us consider that we are interested in the WCRT of a message m_i from class B. Given the qth instance in the busy period, we now compute the $w_i(q)$ as the longest time period from the start of the busy period until the beginning of the transmission of the qth instance. Using this, we may compute the response time of any instance $q \in [1...q_{max}]$ in the busy period and then retain the maximum as the WCRT. This is formalized below and the bound q_{max} will be discussed later in this section.

$$\begin{aligned} w_i(q) = & \max_{m_j \in lp(\text{class } B)} \{C_j\} + (q-1)C_i\left(1 + \frac{\alpha_B^-}{\alpha_B^+}\right) \\ & + \sum_{m_j \in \text{class } B \setminus \{m_i\}} \left\lfloor \frac{(q-1)P_i}{P_j} + 1 \right\rfloor C_j \left(1 + \frac{\alpha_B^-}{\alpha_B^+}\right) \\ & + \sum_{m_j \in hp(\text{class } B)} \left\lfloor \frac{w_i(q)}{P_j} + 1 \right\rfloor C_j, \end{aligned} \tag{20}$$

$$R_i^{\mathrm{w}} = \max_{q=1...q^{\max}} \left\{ w_i(q) - (q-1)P_i + C_i\left(1 + \frac{\alpha_B^-}{\alpha_B^+}\right)\right\}.$$

The first term is the interference due to lower priority message ($lp(\text{class } B)$) and is similar to the analysis for class A. The second term includes the time for the transmission of the $q-1$ instances of m_i as well as the blocking due to the traffic shaper for these messages. The third term includes the transmission time of messages of same priority, i.e., all messages in class B (excluding all instances of message m_i) as well as the blocking due to the traffic shaper for these messages. The forth term includes the interference from higher priority messages ($hp(\text{class } B)$).

The response time of instance q is then computed by adding the (i) time the qth instance waited (relative to its arrival at $(q-1)P_i$) and (ii) its transmission time and the blocking time. The smallest positive q that satisfies the following inequality is indicated by $q^{\max}$,

$$\max_{m_j \in lp(\text{class } B)} \{C_j\} + \sum_{m_j \in \text{class } B} \left\lfloor \frac{(q-1)P_i}{P_j} + 1 \right\rfloor C_j \left(1 + \frac{\alpha_B^-}{\alpha_B^+}\right) + \sum_{m_j \in hp(\text{class } B)} \left\lceil \frac{w_i(q)}{P_j} \right\rceil C_j \leq qP_i.$$

A. Discussion

We shall now discuss Equation 20 in more details with three specific observations. The first observation mentions its relation to the analysis of the CAN protocol [12]. The next two observations are about two key insights that enable us to guarantee the safety of the computed WCRT.

First, as discussed in Section IV-D, the effect of traffic shaper appears as a factor of $\left(1 + \frac{\alpha_B^-}{\alpha_B^+}\right)$ for the transmission times in non-final phases. It has been also discussed that the response time is defined to account for the credit replenishment of messages in the final phase. Therefore, in the worst-case, the transmission time of all messages in class B may be inflated by a factor of $\left(1 + \frac{\alpha_B^-}{\alpha_B^+}\right)$. Once the transmission time of each message is inflated by the discussed factor, the traffic shaper of the class could be ignored and, as we proved in Section IV-D, it is sufficient to consider only one lower priority blocking. Hence, using this transformation, the analysis of an idling protocol looks similar to CAN protocol (which is non-idling) and busy period scenario, except for the fact that we account for the FIFO queue in the class of the message under analysis. However, the major insight is that the traffic shaper and FIFO queue for the higher priority classes may be safely ignored, and this is discussed in the following two observations.

Second, it may be assumed that there does not exist a traffic shaper for *the high priority class*. This assumption is also safe, i.e., it leads to an upper bound for worst-case response time. This is because considering the traffic shaper can only postpone the transmission of messages (in higher priority class) that can potentially lead to moving the transmission of the higher priority messages outside the busy window.

To show this, let us assume that the traffic shaper postpones the transmission of message m_A in the high priority class A. In the busy period scenario, either the transmission of this message is moved outside the busy period interval, or it is still interfering with a message m_B in the low priority class B. In the former case, it is clear that the worst-case response time of messages may not increase by considering the shaper. Therefore, let us focus on the latter. Ignoring the shaper of the higher priority class A, suppose message m_A interferes with the i^{th} instance of message m_B, i.e., $m_B(i)$ and, of course, contributes also to all $m_B(k)$ in the busy period, where $k \geq i$. Since the shaper only postpones the transmission of m_A, once considering the shaper, it interferes with an instance $m_B(j)$, where $i \leq j$ and also contributes to all $m_B(k)$ in the busy period, where $k \geq j$. As it can be observed, once we consider the shaper, higher priority message m_A does not contribute to the response time of $m_B(k)$, where $i \leq k < j$, aside from the fact that the length of the busy period might decrease. In short, this means that instance $m_B(k)$, with $k < i$ or $k \geq j$ is not affected, whereas ignoring the shaper, instance $m_B(k)$, with $i \leq k < j$, experiences also the interference from high priority message m_A. Hence, ignoring the shaper of the higher priority class only leads to a more pessimistic analysis and therefore is safe.

Third, ignoring the FIFO queue for *the high priority class* is safe once the traffic shaper (of the higher priority class) is ignored. The reason is that all the higher priority messages released are sent before the low priority messages (because the shaper is ignored for high priority classes). Since the amount of interference from higher priority (and not the order of the higher priority messages) is the important factor in computing the response time of a low priority message, ignoring the FIFO queues of higher priority classes is safe.

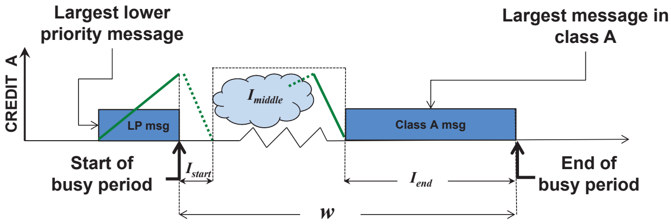

Fig. 6. Approach to improve pessimism with regards to the impact of the traffic shaper on interferences from higher priority messages.

B. A Tighter Analysis

We propose two improvements to the worst-case response time analysis for class B messages that significantly reduce pessimism in the analysis.

Improvement I: Equation 20, while safe, gives pessimistic results because it ignores the fact that the traffic shaper may block some messages from interfering. In fact, the traffic shaper of the higher priority class (class A) may potentially block its messages from interfering with messages of lower priority in class B. We show that this pessimism can be improved. Specifically, we bound the maximum interference allowed by the traffic shaper of a higher priority message and towards this, we consider its interference separately at the start, the middle and the end of the busy period. This is illustrated in Figure 6 that shows the interferences I_{start}, I_{middle} and I_{end}, each of which is described below.

To maximize the interference, let us assume that credit of class A is maximal at the start of the busy period. This occurs when the largest possible lower priority message has blocked class A messages just until the start of the busy period (Figure 6). Note that this scenario bounds the highest possible positive credit for class A and this is given by $\max_{m_j \in lp(\text{class A})} \{C_j\} \, \alpha_A^+$. This upper bound on the credit holds because a lower priority message may block at most once and as soon as the credit is positive the messages in class A will be transmitted. The time interval messages can be transferred using this positive credit is denoted by

$$I_{start} = \max_{m_j \in lp(\text{class A})} \{C_j\} \frac{\alpha_A^+}{\alpha_A^-}.$$

Let us now consider the end of the busy period. For the *worst-case* scenario, we must consider that (i) the last message of class A transmitted within the busy period is the largest and (ii) the credit of class A is just about to be exhausted when the last message of class A arrives. Figure 6 illustrates this scenario. We denote the transmission time taken by this message with

$$I_{end} = \max_{m_j \in \text{class } A} \{C_j\}.$$

Finally, if the busy period is of length w, the remaining time in the middle of the busy period is $w - I_{end} - I_{start}$. Observe that this interval actually starts and ends with zero credits and Corollary 2 can be applied. This implies that the interference in this time interval might be limited by the traffic shaper (see the figure). The maximum interference allowed by the traffic shaper during this time

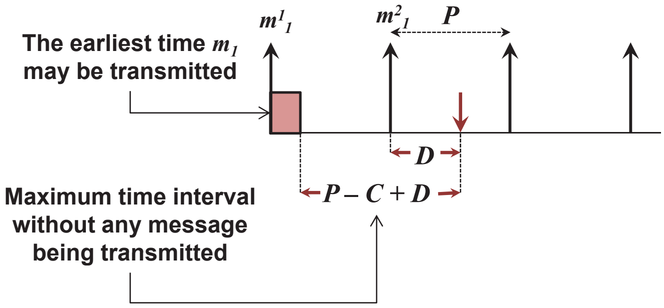

Fig. 7. Quantifying the minimum guaranteed influence from higher priority messages.

interval, based on Corollary 2 in Section III, may be computed as

$$I_{middle}(w) = \max\{0, w - I_{end} - I_{start}\} \frac{\alpha_A^+}{\alpha_A^+ + \alpha_A^-}.$$

The total interference is then given by

$$I_{shaper}(w) = I_{start} + I_{middle}(w) + I_{end}.$$

Note that if the transfer times and periods of the messages in class A are such that they are never blocked by the traffic shaper, the interference due to class A is exactly the same as in Equation 20. In other words, the maximum interference from class A experienced by the low priority class B is the minimum of the maximum demand in class A (i.e., $\sum_{m_j \in \text{class } A} \left\lfloor \frac{w}{P_j} + 1 \right\rfloor C_j$) and the maximum that is allowed to be transmitted by the traffic shaper of class A (i.e., $I_{shaper}(w)$), that is,

$$I_{\text{I}}(w) = \min\left\{ I_{shaper}(w), \sum_{m_j \in \text{class } A} \left\lfloor \frac{w}{P_j} + 1 \right\rfloor C_j \right\}.$$

Improvement II: When a class A (higher priority) is blocking messages of class B (lower priority), it may possibly overlap with the credit recovery of messages in class B. Thus, adding them always, like in Equation 20 gives pessimistic results.

The credit recovery time for class B is $I_B(q)\frac{\alpha_B^-}{\alpha_B^+}$, where $I_B(q)$ is the time units consumed by transmission of the messages of class B within the busy period of length w (For simplicity of presentation, w is used instead of $w(q)$). $I_B(q)$ is formally defined as follows,

$$I_B(q) = \sum_{m_j \in \text{class } B} \left\lfloor \frac{(q-1)P_i}{P_j} + 1 \right\rfloor C_j.$$

Within the the busy period w, we know that at least

$$I_A^{min}(w) = \sum_{m_j \in hp(\text{class B})} \max\left\{0, \left\lceil \frac{w - (P_j - C_j + D_j)}{P_j} \right\rceil \right\} C_j$$

is due to the interference by higher priority messages (see Figure 7). During $I_A^{min}(w)$ (i.e., a lower bound on the load transmitted in the interval of length w), not only is the credit replenishing, but also some load is transmitted. However, in Equation 20, we account for this factor twice, i.e., once in the credit recovery and once in the higher priority interference.

The time that is *exclusively* needed for credit replenishment in an interval of length w is the time for recovering from message transmission in the same class (i.e., $I_B(q)\frac{\alpha_B^-}{\alpha_B^+}$) minus the time for the transmission of higher priority interference, contributing also to credit recovery (i.e., $I_A^{min}(w)$). Thus, the time needed for the credit recovery excluding the overlapping time with interference by higher priority messages, is

$$I_{\text{II}}(w, q) = \max\left\{0, I_B(q)\frac{\alpha_B^-}{\alpha_B^+} - I_A^{min}(w)\right\}.$$

The maximum function indicates the fact that the time for credit recovery cannot be negative.

Improved Analysis: Incorporating the two improvements discussed alone, the following tighter analysis can be used instead of Equation 20,

$$\begin{aligned} w_i(q) = &\max_{m_j \in lp(\text{class } B)} \{C_j\} + (q-1)C_i \\ &+ \sum_{m_j \in \text{class } B \setminus \{m_i\}} \left\lfloor \frac{(q-1)P_i}{P_j} + 1 \right\rfloor C_j \\ &+ I_{\text{I}}(w_i(q)) + I_{\text{II}}(w_i(q), q), \end{aligned} \quad (21)$$

$$R_i^{\text{w}} = \max_{q=1 \ldots q^{\max}} \left\{ w_i(q) - (q-1)P_i + C_i\left(1 + \frac{\alpha_B^-}{\alpha_B^+}\right)\right\}.$$

Similar to previous analyses for fixed-priority scheduling policy, the fixed-point iteration needs to be used to obtain the worst-case response time. Typically, the fixed-point iteration converges to a fixed point (as long as there exists a fixed point) if the monotonicity property holds true. In the proposed analysis, due to improvement II, the length of the busy period, $w(q)$, is not monotonically increasing. However, a decrease (from $w^n(q)$ to $w^{(n+1)}(q)$, where $w^n(q) > w^{(n+1)}(q)$) in the length of the busy period in fixed-point iteration indicates that the resources in the interval $w^n(q)$ are enough to accommodate the messages released in the same interval. Hence, the fixed-point iteration can be stopped immediately when a decrease in the length of the busy period interval is observed. However, as a final technique to reduce the pessimism in the results, we propose to continue iterating using a heuristic based on binary search, to obtain better results. Essentially, the search continues between the last stopped point before the decrease and point with the decrease. We omit a detailed discussion owing to its simplicity.

VII. Experimental Results

In this paper, apart from several formal results, we discussed a basic response-time analysis (Equation 20) and two techniques to reduce the pessimism of this analysis (Equation 21). In this section, we evaluate the improved analysis (Equation 21) against the basic analysis without the improvements (Equation 20). We generated 1000 benchmarks as described in the following. We considered that there are two classes, class A and class B, that are shaped by a traffic shaper. We also assume that there are

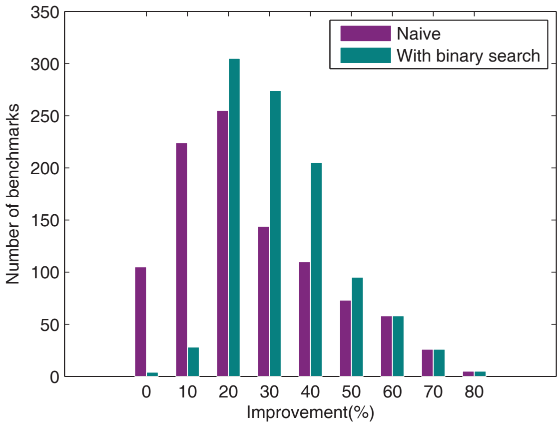

Fig. 8. The improvement (reduction in pessimism) of our proposed analysis (Equation 21) compared to the basic analysis (Equation 20) for the case with naive convergence (purple) and the case with intelligent convergence (green).

lower priority classes whose messages are not influenced by a shaping algorithm. The number of messages in each class was varied randomly between 10 and 20. The send (α^-) and idle (α^+) slopes of the traffic shapers were randomly and independently varied in the interval of $\alpha^-, \alpha^+ \in [1, 2]$. The benchmarks are generated to satisfy the necessary conditions (Equation 3) in each class.

The histogram of the results, plotting the improvements obtained, is shown in Figure 8. Each bar in the histogram shows the number of cases on the y-axis that have the corresponding improvement indicated on the x-axis. The purpule bars are the results related to naive termination, whereas the green bars show the results of a simple heuristic used to continue iterating when we have convergence problem. The metric on the x-axis is relative improvement $\frac{R^{\text{w}}_{\text{bas}} - R^{\text{w}}_{\text{imp}}}{R^{\text{w}}_{\text{bas}}} \times 100$, where $R^{\text{w}}_{\text{bas}}$ and $R^{\text{w}}_{\text{imp}}$ are the worst-case response times found according to the basic (Equation 20) and improved (Equation 21) results, respectively. For instance, we obtained 10% improvement for more than 200 benchmarks, 20% improvement for more than 250 benchmarks and so on. Overall, we obtained improvements of more than 10% for 900 out of 1000 benchmarks. Also, as it can be observed, the maximum improvement achieved by our analysis can be as large as 83%, while the average improvement achieved is 25%.

These results in Figure 8 are reported for the case where the iterations on the busy period were stopped in a naive fashion (i.e., stop iteration once the convergence problem is observed), as reported in Section VI-B. The results of the case when we continue iterating with our proposed method that goes beyond convergence, to obtain safe but more accurate results, are also shown in Figure 8. While the maximum improvement is still the same, the average improvement achieved is improved to 33%. Moreover, we observe that the number of cases corresponding to 0% and 10% improvement are drastically decreased compared to the naive case. Instead the number of benchmarks where we achieve higher improvements of 30% to 50% have significantly gone up.

We measure the runtime of the two approaches on a PC with a quad-core CPU running at 2.83 GHz with 8 GB of RAM and Linux operating system. The average runtime of analysis with the naive termination is 2.54 ms, whereas the average runtime of the analysis that uses a heuristic based on binary search is increased to 2.79 ms.

VIII. Conclusion

In this work, we provided a formal basis for Ethernet AVB schedulability analysis apart from providing utilization based bounds on necessary conditions. We also proposed techniques to reduce the pessimism in the WCRT analysis for Ethernet AVB and our experiments show significant improvements. As noted, almost all our results may be readily generalized to the case where more than 2 classes have a traffic shaper.

References

[1] L. Lo Bello, "The case for Ethernet in automotive communications," *SIGBED Review*, vol. 8, no. 4, Dec. 2011.

[2] M. Di Natale, H. Zeng, P. Giusto, and A. Ghosal, *Understanding and Using the Controller Area Network Communication Protocol*. Springer, 2012.

[3] J. Diemer, D. Thiele, and R. Ernst, "Formal worst-case timing analysis of Ethernet topologies with strict-priority and AVB switching," in *International Symposium on Industrial Embedded Systems*, 2012.

[4] F. Reimann, S. Graf, F. Streit, M. Glaß, and J. Teich, "Timing analysis of Ethernet AVB-based automotive E/E architectures," in *International Conference on Emerging Technology & Factory Automation*, 2013.

[5] J. Georges, T. Divoux, and E. Rondeau, "Strict priority versus weighted fair queueing in switched Ethernet networks for time critical applications," in *International Parallel and Distributed Processing Symposium*, 2005.

[6] J. Imtiaz, J. Jasperneite, and L. Han, "A performance study of Ethernet Audio Video Bridging (AVB) for industrial real-time communication," in *Emerging Technologies Factory Automation, 2009*, 2009.

[7] K. C. Lee, S. Lee, and M. Lee, "Worst case communication delay of real-time industrial switched Ethernet with multiple levels," *Industrial Electronics, IEEE Transactions on*, vol. 53, no. 5, 2006.

[8] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, "Trends in automotive communication systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204–1223, June 2005.

[9] G. Alderisi, G. Patti, and L. Lo Bello, "Introducing support for scheduled traffic over IEEE Audio Video Bridging networks," in *Emerging Technologies and Factory Automation*, 2013.

[10] D. Thiele, J. Diemer, P. Axer, R. Ernst, and J. Seyler, "Improved formal worst-case timing analysis of weighted round robin scheduling for Ethernet," in *Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2013.

[11] H. Bauer, J. Scharbarg, and C. Fraboul, "Worst-case backlog evaluation of avionics switched ethernet networks with the trajectory approach," in *Euromicro Conference on Real-Time Systems (ECRTS)*, 2012.

[12] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007.

[13] K. Tindell and J. Clark, "Holistic schedulability analysis for distributed real-time systems," *Euromicro Journal on Microprocessing and Microprogramming (Special Issue on Parallel Embedded Real-Time Systems)*, vol. 40, pp. 117–134, 1994.

[14] J. C. Palencia Gutiérrez and M. González Harbour, "Schedulability analysis for tasks with static and dynamic offsets," in *Real-Time Systems Symposium*, 1998.

[15] R. Davis, S. Kollmann, V. Pollex, and F. Slomka, "Controller area network (CAN) schedulability analysis with FIFO queues," in *Euromicro Conference on Real-Time Systems*, 2011.

[16] J. Lehoczky, "Fixed priority scheduling of periodic task sets with arbitrary deadlines," in *Real-Time Systems Symposium*, 1990.