Real-Time Capable CAN to AVB Ethernet Gateway Using Frame Aggregation and Scheduling

Christian Herber, Andre Richter, Thomas Wild, Andreas Herkersdorf

Technische Universität München - Institute for Integrated Systems
Munich, Germany
christian.herber@tum.de, andre.richter@tum.de, thomas.wild@tum.de, herkersdorf@tum.de

***Abstract*—Ethernet is a key technology to satisfy the communication requirements of future automotive embedded systems. Audio/Video Bridging (AVB) Ethernet is a set of IEEE standards that allows synchronous and time-sensitive communication. It is the favored candidate for backbone and camera applications, but is not expected to replace Controller Area Network (CAN). Instead, both have to coexist in future architectures. No research has been conducted regarding CAN to AVB gateways, and approaches for similar protocols are either not fit or inefficient.**

In this paper, we present a CAN to AVB Ethernet gateway that allows efficient, real-time capable forwarding. We aggregate and schedule multiple CAN frames into a single AVB Ethernet frame to minimize bandwidth requirements. We evaluate static and dynamic scheduling approaches and determine optimal gateway configurations, showing that the necessary bandwidth reservation is reduced by 72% compared to similar approaches.

***Index Terms*—Audio/Video Bridging, Controller Area Network, automotive electronics, gateway.**

I. Introduction

Today's automotive embedded systems are composed of up to 100 electronic control units (ECUs) and interconnected by a variety of fieldbuses, including CAN, FlexRay, MOST, and LIN. ECUs are distributed throughout the car and implement dedicated electronic functions. They are grouped into functional domains like body, infotainment, chassis, powertrain etc. and are connected through a central gateway.

Recognizing the limited scalability of this approach, car manufacturers are transitioning into domain controlled architectures [1], [2]. Here, central domain controllers consolidate multiple previously distributed functions, while decentral slave nodes persist to access sensors and actuators. Domain controllers communicate through a backbone network [3].

A central backbone requires high bandwidth and real-time capability at the same time. None of the currently employed interconnects satisfies these requirements. BroadR-Reach, a recently introduced physical layer standard reaches 100 Mbit/s Ethernet communication using unshielded twisted pair cable [3]. Legacy Ethernet has limited ability to provide end-to-end latency guarantees [4], which makes AVB Ethernet the favored candidate for such backbone applications.

The transition to real-time Ethernet is not expected to be a revolution, but rather a slow evolutionary process, because of the high amount of legacy components used in the automotive domain [5]. Therefore, existing fieldbuses will coexist with AVB Ethernet. Due to low cost, high reliability, and wide adoption, CAN is assumed to remain an essential part in automotive electronics long after the introduction of Ethernet.

The coexistence of AVB Ethernet and legacy protocols like CAN requires efficient forwarding between different interconnect technologies. In this paper, we propose a mechanism for forwarding time-sensitive CAN frames to AVB Ethernet networks. Our work transfers and extends related work (Section III) regarding CAN to Ethernet gateways considering AVB specific characteristics. Our gateway concept (Section IV) aggregates multiple CAN frames into a larger AVB Ethernet frame to reduce the framing overhead. Additionally, we analyze message scheduling approaches with respect to their ability of providing real-time capable forwarding. We consider first in, first out (FIFO) ordering as well as static and dynamic prioritization. Using schedulability analysis (Section V), we show that we could achieve real-time capability with 72% less reserved bandwidth than current approaches.

II. Background

In this section, we introduce fundamental operating principles and notations regarding CAN and AVB Ethernet and compare them with respect to their framing overhead.

A. Controller Area Network

Controller Area Network (CAN) is the most prevalent bus in automotive embedded systems. It provides robust real-time communication at low cost. CAN is assumed to remain an important part of automotive architectures even after the introduction of real-time capable Ethernet.

Because CAN does not have a concept of sender/receiver addresses, messages are always broadcasted. The content of a message can be derived from its ID. The bus access scheme follows non-preemptive strict-priority scheduling, using message IDs as priority. We denote the set of messages which partake in the communication of a CAN bus as $\mathcal{M} = \{m_0, ..., m_{M-1}\}$. Each message is associated with a payload length s_m and a minimum inter-arrival time T_m.

The response time of each message describes the time from release until its successful transmission. The best-case response time is equivalent to the minimum transmission time C_m^{min}. The worst-case response time (WCRT) R_m can be calculated analytically as described in [6].

This work was funded within the project ARAMiS by the German Federal Ministry for Education and Research with the funding IDs 01|S11035. The responsibility for the content remains with the authors.

The frame length in CAN can differ for multiple instances of the same message, depending on its contents. CAN does not carry an explicit clock signal and nodes synchronize based on sender bit timings. Therefore, additional stuff bits are added after five bits of equal polarity. According the definition of CAN data frames [7], the maximum (assuming maximum bit-stuffing) and minimum transmission time C_m^{max} and C_m^{min} can be calculated as

$$C_m^{max} = L_m^{max}\tau_{can} = (55 + 10s_m)\tau_{can}, \quad (1)$$

$$C_m^{min} = L_m^{min}\tau_{can} = (47 + 8s_m)\tau_{can}, \quad (2)$$

where $L_m^{max/min}$ denotes maximum/minimum frame length and τ_{can} is the bit time on the bus. CAN is specified to operate with up to 1 Mbit/s, but practical implementations are limited to 500 kbit/s (equivalent to $\tau_{can} = 2\mu s$).

B. AVB Ethernet

AVB Ethernet is specified by a set of IEEE standards. It provides time-synchronous and real-time capable communication over full-duplex switched Ethernet. While originally developed for audio and video applications, it has attracted attention from the automotive industry, where increasing bandwidth requirements result in the need to introduce new interconnect technologies.

Traffic is grouped into multiple traffic classes, which can carry either real-time or legacy Ethernet traffic. Forwarding of traffic follows a strict class-based priority scheme. Traffic within real-time capable classes is policed using a credit-based shaper algorithm. This shaping is employed in all senders and switches throughout the network.

Senders can register streams across the network. Each stream is associated with a traffic class and has a fixed reserved bandwidth and maximum frame length. From the topology and configuration of all streams, end-to-end latencies can be calculated analytically [8].

Additionally, AVB Ethernet offers time synchronization following IEEE 802.1AS [9]. Because CAN is an asynchronous protocol, it does not play a role in our design.

Encapsulation of legacy control frames in time-sensitive streams is defined in IEEE P1722a. Specifically, it defines a payload format to transport CAN frames via AVB Ethernet. Fig. 1 illustrates, how multiple CAN frames can be encapsulated in an AVB Ethernet frame. The payload of VLAN capable Ethernet frames starts with a control stream header, which can be followed by multiple CAN messages. The *CAN MSG Info* field contains header information like CAN ID, bus ID, payload length etc. *Info* and *Payload* field are both 64 bit.

The length of an AVB Ethernet frame encapsulating N_{can} CAN frames is

$$L_{avb} = (336 + 128N_{can})\ bit. \quad (3)$$

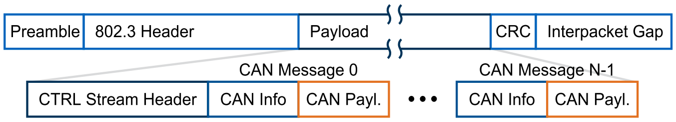

Fig. 1. Encapsulation of CAN frames in AVB Ethernet Frames

The high framing overhead in AVB Ethernet compared to CAN requires multiple CAN frames to be encapsulated for efficient implementations.

III. Related Work

CAN to Ethernet gateways have been proposed for legacy [10] and Avionics Full Duplex Switched (AFDX) [11] Ethernet, a real-time capable network technology used in avionics. Both use frame aggregation mechanisms. Gateways for AVB Ethernet have only been proposed for FlexRay and MOST [5].

Kern et al. [10] optimized a CAN to legacy Ethernet gateway to provide low forwarding latencies and small framing overhead. A buffer for CAN messages within the gateway is completely forwarded when the buffer is full, a timeout occurs, or upon arrival of a high priority message at the gateway. The outgoing Ethernet traffic pattern can be bursty, with multiple frames being sent back-to-back. Due to stream traffic shaping, this is not possible in AVB. Frames would queued within the stream buffer, which in consequence makes the scheme equivalent to FIFO scheduling.

Ayed et al. [11] proposed a similar strategy under consideration of AFDX specific characteristics. Communication happens across virtual links (VLs), which have a reserved bandwidth and maximum frame size. The gateway releases Ethernet frames periodically. The maximum frame size is chosen to fit the maximum amount of received CAN payloads in one period, resulting in significant overreservation. Schedulability analysis is used to evaluate the real-time capability.

Both approaches are designed to completely release all buffered CAN messages at a time using a single Ethernet frame. Therefore, no scheduling among the CAN messages is necessary. AVB is similar to AFDX, as it uses maximum frame sizes and fixed reserved bandwidth (streams instead of VLs). The concepts from [11] can therefore be transferred. We will refer to this concept as complete release (CR) and will use it as comparison throughout our evaluation.

Integration of multiple data units into one transmission unit has been applied in other areas. IEEE 802.11n wireless LAN uses frame aggregation techniques to reduce average framing overheads and increase overall throughput [12]. A frame aggregation scheduler has been proposed in [13]. It prioritizes packets based on their expiration time and releases the aggregated packets to minimize the average drop rate. An optimal frame size is determined dynamically based on the bit error rate of the channel.

A similar principle is used to reduce the computational load imposed from TCP/IP processing for 10 Gbit/s Ethernet. Receive side coalescing [14] is an extension to Intel's network interface controllers, in which packets belonging to the same TCP/IP flow are concatenated into one larger packet.

In this work, we present a CAN to AVB Ethernet gateway with real-time capable forwarding. We use a frame aggregation mechanism to reduce the framing overhead. In contrast to concepts proposed for AFDX Ethernet [11], we use scheduling to improve the efficiency of the gateway.

IV. Gateway Forwarding Strategy

The gateway has to act as an interface between a CAN bus and an AVB Ethernet network. These protocols are mismatched in framing overhead, payload length, available bandwidth, arbitration and network structure. In this paper, we focus on forwarding from CAN to AVB Ethernet, because the opposite direction leaves little design options: CAN frames encapsulated in AVB frames separately enter CAN arbitration. Also, our real-time analysis focuses on CAN and gateway induced latencies. AVB network latencies are not part of this analysis and have to be considered additionally in a system wide analysis. We formulate the following design goals:

1) **Real-time capability:** Each message must be transmitted on the CAN bus and forwarded to the AVB network before its deadline D_m. We assume implicit deadlines ($D_m = T_m$). The overall system is schedulable, if the time from message release till successful forwarding is smaller than the deadline for all messages.
2) **Efficiency:** Schedulability should be achieved using minimal resources. Specifically, CAN over AVB streams should require minimal bandwidth. The bandwidth of a stream directly affects fan-in delays experienced by same priority streams and blocking of lower priority streams [8].

Data transmission across AVB networks happens in the form of streams, which have fixed reserved bandwidths. Traffic shaping for streams within every endpoint forces an essentially cyclic sending behavior. We define the interval between two CAN over AVB frames as T_{avb}.

Because real-time capability is hard to achieve in fully utilized systems, we allow overreservation, i.e. there is more bandwidth available than absolutely necessary. We introduce an overreservation factor OR, which decreases the interval T_{avb} between two CAN over AVB frames.

$$T_{avb}(OR) = T_{avb}(0) \,/\, (1 + OR) \,. \tag{4}$$

For example, $OR = 100\%$ means available resources are doubled. We derive the sending interval without overreservation $T_{avb}(0)$ depending on the number N_{can} of CAN frames encapsulated in a single AVB Ethernet frame as

$$T_{avb}(0) = N_{can} / \sum_{\forall k \in \mathcal{M}_{fwd}} \frac{1}{T_k} . \tag{5}$$

The bandwidth reserved for the CAN over AVB stream is

$$bw_{avb,res}(N_{can}, OR) = L_{avb} / T_{avb} . \tag{6}$$

Overreservation OR increases reserved bandwidth, but at the same time reduces forwarding latencies, because frames are sent more often. On the other hand, increasing N_{can} at a constant level of OR reduces the required bandwidth, but leads to increased latencies. Therefore, modifying either parameter improves the gateway with respect to one of the design goals and worsens it with respect to the other. Finding the optimal configuration is a non-trivial task that depends on the scheduling strategy for frame forwarding. Following, we will analyze three possible schedulers including FIFO, strict priority and earliest deadline first. Section V evaluates them in their ability to satisfy the design goals for different combinations of N_{can} and OR.

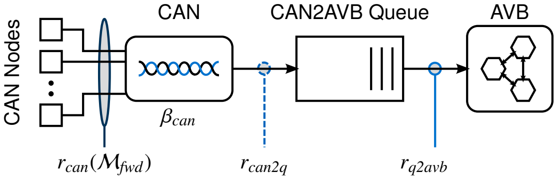

Fig. 2. Queuing model of the CAN-AVB gateway

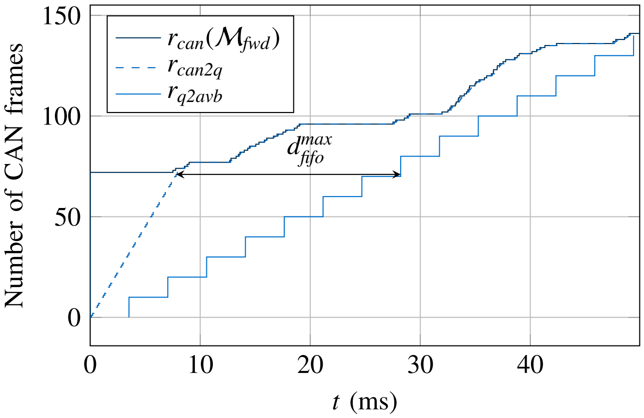

Fig. 3. FIFO forwarding: $N_{can} = 10$, $OR = 50\%$

A. FIFO Forwarding

Messages arriving from the CAN bus have to be buffered within the gateway. Even when using overreservation for the outgoing stream, it may not be possible to quickly forward all designated messages arriving in short succession. In this section, we will investigate a FIFO forwarding scheme.

FIFO scheduling is appealing because of its simplicity. Arriving messages are added to the end of the queue. When a CAN over AVB frame is about to be sent, N_{can} frames are taken from the front of the queue and transmitted. Following, we present an analytic method to calculate worst-case latency bounds.

The analysis is based on the queuing model depicted in Fig. 2. CAN nodes release messages towards the CAN bus where they are arbitrated using strict priorities. After successful transmission, messages arrive at the gateway and are enqueued. Messages are dequeued periodically and send to the AVB network. The analysis uses network calculus [15] to compute the patterns of arriving and departing traffic. From this information, delay bounds can be derived.

To cover the worst-case for the arrival of messages at the gateway queue, we first consider the cumulative amount of messages that are queued for transmission on the CAN bus within the interval $[0, t)$. This includes all message instances, which are released or transmitted during this interval. It can be calculated as

$$r_{can}(\mathcal{M}_{fwd}, t) = \sum_{\forall k \in \mathcal{M}_{fwd}} \lceil (t + R_k) / T_k \rceil . \tag{7}$$

The equation assumes a cyclic release of messages. It also considers that the inter-arrival time of two message instances can be reduced by up to the WCRT R_m, if the first message instance was delayed in a worst-case scenario.

The arrival of messages at the gateway queue is constraint by the limited transmission speed of the CAN bus. The minimum spacing between messages is equivalent to the minimum transmission time of a CAN message. The maximum amount of message arrivals in the interval $[0, t)$ is given as

$$\beta_{can}(t) = \lceil t/C_{min} \rceil . \tag{8}$$

By combining (7) and (8) using min-plus convolution [15], we calculate an upper bound for the cumulative arrival of messages at the gateway queue

$$\begin{aligned} r_{can2q}(t) &= \left(r_{can}\left(\mathcal{M}_{fwd}\right) \otimes \beta_{can}\right)(t) \qquad (9) \\ &= \inf_{0 \le \tau \le t} \left\{ r_{can}\left(\mathcal{M}_{fwd}, \tau\right) + \beta_{can}\left(t - \tau\right) \right\}. \qquad (10) \end{aligned}$$

On the other end of the queue, N_{can} messages are removed cyclic in an interval of T_{avb} and transmitted across the AVB network. In a worst-case scenario, the last frame has been sent an infinitesimal time before $t = 0$. The cumulative number of messages removed is

$$r_{q2avb}(t) = \lfloor t/T_{avb} \rfloor N_{can}. \tag{11}$$

Equations (7), (9), and (11) are visualized in Fig. 3 for an exemplary configuration. In this configuration, 50% of messages from a CAN bus with a bandwidth of 500 kbit/s are forwarded to the AVB network. The same scenario will be used for visualization of the other scheduling approaches.

The vertical discrepancy between r_{can2q} and r_{q2avb} in Fig. 3 translates to a backlog. We derive the queueing delay from the horizontal derivation between cumulative inflow and outflow of messages. Therefore, a message enqueued at time t is guaranteed to be forwarded after

$$d_{fifo}(t) = \inf_{\forall t \ge 0} \left\{ t_d : r_{can2q}\left(t\right) - r_{q2avb}\left(t + t_d\right) \le 0 \right\}. \tag{12}$$

The maximum queuing delay for FIFO forwarding can therefore be bounded as

$$d_{fifo}^{max} = \sup_{\forall t \ge 0} \left\{ d_{fifo}(t) \right\}. \tag{13}$$

FIFO forwarding is easy to implement and capable of guaranteeing latency bounds for the forwarding operation. However, the latency bounds are equal for low and high priority messages, which can differ in end-to-end latency requirements in orders of magnitude. Thus, we will propose prioritized forwarding mechanisms in the next sections.

B. Strict Priority (SP) Forwarding

In SP forwarding, each message is assigned a static priority at design time. Messages arriving at the gateway are enqueued in a priority sorted list. When a CAN over AVB frame is assembled, the N_{can} highest priority messages are transmitted and removed.

In contrast to FIFO forwarding, the order of departing frames can differ from the incoming order. We assume that the forwarding of a message can be delayed by up to $d_{m,sp}$. Further, we derive the worst-case amount of higher priority message instances which block the forwarding of message m as

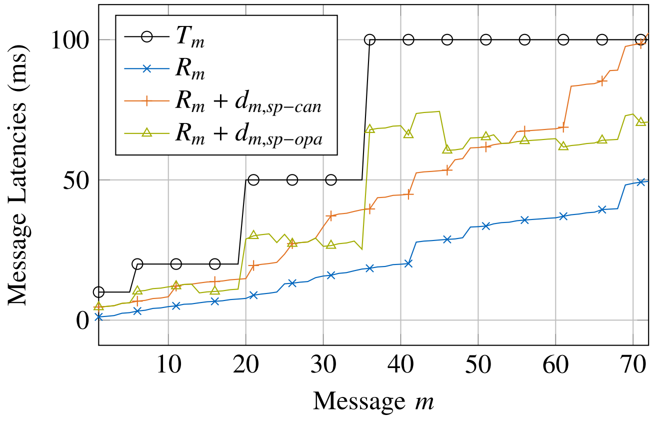

Fig. 4. Strict priority forwarding delays with CAN and optimal (OPA) priorities, respectively: $N_{can} = 10$, $OR = 50\%$

$$I_m = \sum_{\forall k \in hp(m)} \left\lceil \frac{d_{m,sp} + R_k}{T_k} \right\rceil, \tag{14}$$

with $hp(m) \subset \mathcal{M}_{fwd}$ being the set of forwarded higher priority messages. This equation is similar to (7), as it makes the same assumptions on arriving CAN messages.

A message m can only be forwarded, when no higher priority messages reside in the queue. Assuming that a CAN over AVB frame has just been released when message m arrives at the gateway, the queueing delay is given by the implicit formulation

$$d_{m,sp} = T_{avb}\left(1 + \left\lfloor \frac{I_m(d_{m,sp})}{N_{can}} \right\rfloor\right), \tag{15}$$

which can be solved using (14) and fixed-point iteration.

We use two different methods for priority assignment within the gateway. For one, we use CAN priorities also for forwarding. However, this is suboptimal, because low priority CAN messages are subject to more blocking on the CAN bus and should therefore have adjusted priorities within the gateway.

In preemptive SP systems, "deadline minus jitter" (D–J) monotonic priority assignment is optimal [16]. This is not generally true for non-preemptive systems like CAN if message lengths differ between messages [6]. With our forwarding scheme, blocking time is message length independent, because all messages occupy the same share of an Ethernet frame. Therefore, we consider (D–J) monotonic priorities an optimal priority assignment (OPA).

Messages arrive at the gateway queue and therefore at the SP scheduler somewhen between their minimum transmission time C_m^{min} and their WCRT R_m. Thus, messages arrive with a remaining deadline of $D_m - C_m^{min}$ and a jitter of $R_m - C_m^{min}$. (D–J) monotonic priority assignment therefore translates into giving higher priorities to smaller values of $D_m - R_m$.

Fig. 4 presents the cycle time T_m, the WCRT on the CAN bus R_m and the combined delay of CAN bus and gateway forwarding queuing delays for an example message set using CAN priorities and optimal priorities (OPA), respectively. Assuming implicit deadlines ($D_m = T_m$), we observe a deadline violation for a low priority message using CAN priorites, while OPA guarantees a significant slack for all messages.

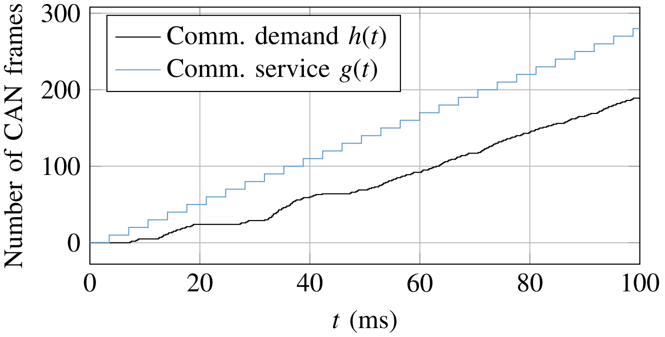

Fig. 5. EDF forwarding schedulability test: $N_{can} = 10$, $OR = 50\%$

C. Earliest Deadline First (EDF) Forwarding

Using EDF as scheduler, CAN frames queued within the gateway are sorted with respect to the remaining time until their deadline. The gateway has no information about how long a message was queued before being successfully transmitted on the CAN bus. Thus, we make the pessimistic assumption that a worst-case occurred and a message m was received R_m after its release.

Baruah et. al [17] showed that a task set is schedulable on a uniprocessor, if the processor demand $h(t)$ is smaller or equal to t. The processor demand is composed of the execution time of all jobs that have a deadline until t. In analogy, we consider a communication demand function $h(t)$. It describes the number of CAN frames which have to be forwarded until t:

$$h(t) = \sum_{\forall m \in \mathcal{M}_{fwd}} \max\left\{0, 1 + \left\lfloor \frac{t - (D_m - R_m)}{T_m} \right\rfloor\right\}. \quad (16)$$

The equation assumes that deadlines D_m are reduced by the WCRT on the CAN bus R_m.

We additionally introduce a communication service function $g(t)$, which describes the number of CAN frames guaranteed to be forwarded at t. It assumes that an Ethernet frame has just been released before $t = 0$. Afterwards, N_{can} CAN frames are forwarded at every multiple of T_{avb}. We consider the system schedulable, if the following condition holds

$$h(t) \leq g(t) = N_{can} \lfloor t/T_{avb} \rfloor . \quad (17)$$

Fig. 5 visualizes the schedulability test. The offered communication service is a step function, where the height of the step is equivalent to the number of CAN frames within an AVB frame. If the demand is lower than the service at all times, schedulability is guaranteed.

V. SCHEDULABILITY EVALUATION

The main design goal of the gateway is timely forwarding of messages before a predefined deadline using minimal resources. Therefore, we measure the schedulability of various configurations in a design space exploration. Following, we present a reproducible scenario and discuss the results.

A. Scenario

We designed the scenario to be as close as possible to realistic automotive settings based on data obtained from in-car measurements [18]. In order to produce general results, we explore a variety of configurations as described below.

We consider four different forwarding techniques including first in, first out (**FIFO**), strict priority based on CAN IDs (**SP-CAN**) and with optimal priority assignment (**SP-OPA**), and earliest deadline first (**EDF**). For comparability, we also considered a forwarding scheme we adapted from [11], to which we will refer to as complete release (**CR**). It is equivalent to a FIFO configuration, in which all messages received within a sending interval T_{avb} can be fit into a CAN over AVB frame.

We vary the number of CAN frames encapsulated in one AVB Ethernet frame in the range of $N_{can} \in \{1, 2, ..., 35\}$. Additionally, we consider overreservation of bandwidth $OR \in \{0, 10, ..., 400\}\%$. We evaluated the gateway with different traffic patterns coming from the CAN bus. CAN messages are sent cyclic with cycle times $T_m \in \{10, 20, 50, 100\}ms$. These cycle times are assigned randomly with probability 4.8%, 14.3%, 33.3% and 47.6%, respectively. The overall utilization of the CAN bus is $U = 80\%$ throughout the evaluation. The available CAN bandwidth is 500 kbit/s. A subset $\mathcal{M}_{fwd} \subset \mathcal{M}$ of all messages (including 50% of the traffic) will be forwarded towards the AVB backbone.

B. Results

We consider a message set schedulable, if all forwarded messages $m \in \mathcal{M}_{fwd}$ are guaranteed to be forwarded before their deadline D_m. To evaluate the performance of each forwarding mechanism in different configurations, we randomly generated and tested 10,000 message sets.

We define the schedulability S for each configuration as the ratio of message sets deemed schedulable using the analyses from Section IV. It is dependent on the scheduling algorithm used for forwarding, the number of CAN frames encapsulated in an AVB Ethernet frame N_{can}, and the overreservation OR.

We consider a configuration of N_{can} and OR optimal, if no configuration achieves a higher level of schedulability using less bandwidth for the respective scheduling algorithm. Fig. 6 presents such optimal configurations for every forwarding mechanism. It is a direct measure to assess a mechanism's ability in efficient and real-time capable forwarding. For reference, we also included schedulability for configurations restricted to $N_{can} = 1$, meaning no frame aggregation is used.

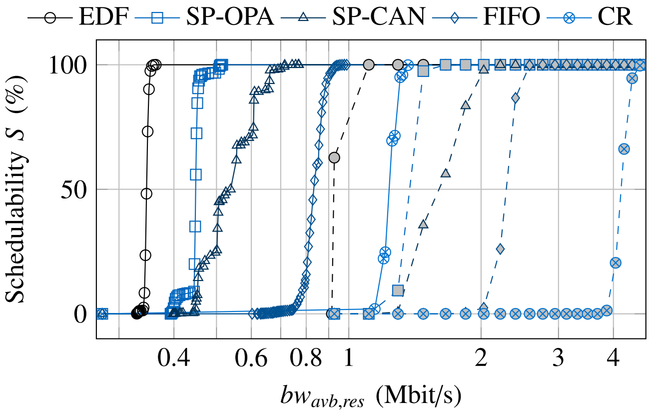

Fig. 6. Schedulability for optimal gateway configurations with frame aggregation (solid lines, $N_{avb} > 1$) and without (dashed lines, $N_{avb} = 1$)

The results clearly indicate that an improvement in schedulability is achieved when we increase the complexity of the scheduler. In particular, EDF gives the best performance for all configurations. This is expected, because EDF is proven to be an optimal schedule [19], i.e. if a configuration is not schedulable using EDF, it is not schedulable at all. Similarly, reordering priorities in an SP scheme improves schedulability.

CR is only applicable in a subset of configurations, in which all buffered CAN frames can always be packed into one outgoing Ethernet frame. Because frames don't have to be scheduled, all approaches would yield the same performance. Using scheduling, additional configurations are available, which explains why even FIFO ordering outperforms CR.

In theory, FIFO could outperform SP in a few unrealistic scenarios, where messages arrive at the gateway with similar deadlines. If $(D_m - R_m)$ is equal for every message, FIFO equals EDF forwarding and would therefore outperform SP schemes. Because automotive latency requirements are usually diverse, the lack of differentiated prioritization leads to bad real-time performance in FIFO forwarding.

For quantitative comparison, we consider the bandwidth reservation necessary to achieve schedulability $S \geq 50\%$. Using a simple FIFO forwarding, we reduced bandwidth requirements by 33.38% compared to CR. With the best and most complex schedule, EDF, a total reduction of 72.21% compared to CR is achieved. Nevertheless, FIFO based forwarding is interesting because of its low implementation cost, e.g. when an hardware offloading is desired.

Real-time analyses are pessimistic in nature, and in this gateway scenario, worst-case scenarios for CAN bus and gateway forwarding are considered. For all scheduling mechanisms, we assume that one message instance can experience both worst-cases during one transmission. Nevertheless, comparability among schedulers is given, as we make the same assumptions for all of them. Also, despite pessimism, we still achieve significant improvements in schedulability.

VI. Conclusion

The increasing bandwidth demand of automotive embedded systems requires new interconnect technologies like AVB Ethernet. To enable the coexistence of AVB Ethernet and the most prevalent legacy interconnect Controller Area Network (CAN), we presented a gateway between these protocols. The gateway design followed two goals. First, the gateway must enable real-time capable forwarding. Second, it should use minimal AVB resources (bandwidth).

To minimize the framing overhead, we used a frame aggregation technique to encapsulate multiple CAN frames within a single AVB Ethernet frame. To reduce forwarding latencies, we allow overreservation of AVB resources. Scheduling algorithms including FIFO, strict priority (SP), and earliest deadline first (EDF) are used to select the next CAN frames to be forwarded via AVB Ethernet. For each forwarding mechanism, we provide an analytic framework to determine the real-time capability of the gateway.

Using automotive CAN patterns, we evaluated a wide range of gateway configurations. As expected, EDF was able to achieve real-time capability using the least resources. We were able to encapsulate around 15 CAN frames in one AVB frame depending on the specific traffic scenario. Compared to previous forwarding mechanisms we adopted from avionics, we reduced the necessary bandwidth reservation by 72.21%.

Our contributions enable efficient forwarding of CAN frames with minimal resource consumption within the AVB network, which allows scalability of future automotive architectures and facilitates incremental design.

References

[1] D. Reinhardt and M. Kucera, "Domain controlled architecture: A new approach for large scale software integrated automotive systems," in *Pervasive and Embedded Computing and Communication Systems*, 2013, pp. 221–226.

[2] M. Di Natale and A. L. Sangiovanni-Vincentelli, "Moving from federated to integrated architectures in automotive: The role of standards, methods and tools," *Proceedings of the IEEE*, vol. 98, no. 4, pp. 603–620, 2010.

[3] P. Hank, S. Müller, O. Vermesan, and J. Van Den Keybus, "Automotive ethernet: in-vehicle networking and smart mobility," in *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 2013, pp. 1735–1739.

[4] H.-T. Lim, L. Völker, and D. Herrscher, "Challenges in a future ip/ethernet-based in-car network for real-time applications," in *Proceedings of the 48th Design Automation Conference*. ACM, 2011, pp. 7–12.

[5] H. Zinner, J. Noebauer, T. Gallner, J. Seitz, and T. Waas, "Application and realization of gateways between conventional automotive and ip/ethernet-based networks," in *Proceedings of the 48th Design Automation Conference*. ACM, 2011, pp. 1–6.

[6] 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.

[7] *CAN Specification 2.0*, Robert Bosch GmbH Std., 1991.

[8] P. Axer, D. Thiele, R. Ernst, and J. Diemer, "Exploiting shaper context to improve performance bounds of ethernet avb networks," in *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference*. ACM, 2014, pp. 1–6.

[9] G. M. Garner and H. Ryu, "Synchronization of audio/video bridging networks using ieee 802.1 as," *Communications Magazine, IEEE*, vol. 49, no. 2, pp. 140–147, 2011.

[10] A. Kern, D. Reinhard, T. Streichert, and J. Teich, "Gateway strategies for embedding of automotive can-frames into ethernet-packets and vice versa," in *Architecture of Computing Systems-ARCS 2011*. Springer, 2011, pp. 259–270.

[11] H. Ayed, A. Mifdaoui, and C. Fraboul, "Frame packing strategy within gateways for multi-cluster avionics embedded networks," in *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*. IEEE, 2012, pp. 1–8.

[12] D. Skordoulis, Q. Ni, H.-H. Chen, A. P. Stephens, C. Liu, and A. Jamalipour, "Ieee 802.11 n mac frame aggregation mechanisms for next-generation high-throughput wlans," *Wireless Communications, IEEE*, vol. 15, no. 1, pp. 40–47, 2008.

[13] T. Selvam and S. Srikanth, "A frame aggregation scheduler for ieee 802.11 n," in *Communications (NCC), 2010 National Conference on*. IEEE, 2010, pp. 1–5.

[14] S. Makineni, R. Iyer, P. Sarangam, D. Newell, L. Zhao, R. Illikkal, and J. Moses, "Receive side coalescing for accelerating tcp/ip processing," in *High Performance Computing 2006*. Springer, 2006, pp. 289–300.

[15] J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. Springer, 2001, vol. 2050.

[16] L. George, N. Rivierre, M. Spuri *et al.*, "Preemptive and non-preemptive real-time uniprocessor scheduling," Institut National de Recherche et Informatique et en Automatique (INRIA), Tech. Rep., 1996.

[17] S. K. Baruah, A. K. Mok, and L. E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor," in *Real-Time Systems Symposium, 1990. Proceedings., 11th*. IEEE, 1990, pp. 182–190.

[18] B. Müller-Rathgeber, M. Eichhorn, and H.-U. Michel, "A unified car-it communication-architecture: Design guidelines and prototypical implementation," in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 709–714.

[19] K. Jeffay, D. F. Stanat, and C. U. Martel, "On non-preemptive scheduling of period and sporadic tasks," in *Real-Time Systems Symposium, 1991. Proceedings., Twelfth*. IEEE, 1991, pp. 129–139.