Deadline-aware Online Scheduling of TSN Flows for Automotive Applications

Gaetano Patti, *Member, IEEE*, Lucia Lo Bello, *Senior Member, IEEE*, Luca Leonardi, *Member, IEEE*.

***Abstract*—The Time-Sensitive Networking (TSN) set of standards allows to support on the same channel the different kinds of traffic flows that are typically found in automotive scenarios. This work proposes the introduction of online Earliest Deadline First-based scheduling in TSN to provide support for event-driven real-time traffic. The proposed approach, called Deadline-TSN, is an online approach and therefore, unlike other approaches in the literature, it does not require complex offline schedule calculations. Moreover, Deadline-TSN is able to uniformly deal with real-time periodic and event-driven traffic flows. The paper presents Deadline-TSN and provides both a worst-case response time analysis and simulative assessments in realistic automotive scenarios.**

***Index Terms*—Time-Sensitive Networking (TSN), IEEE 802.1, Earliest Deadline First (EDF), Real-time networks, Automotive networks.**

I. Introduction

THE IEEE 802.1 Time-Sensitive Networking (TSN) set of standards enables properties, such as time synchronization, reliability and determinism, that are very suitable for industrial and automotive [1] communications over Ethernet links. The TSN standards realize a converged network able to support multiple traffic flows, with different timing and reliability requirements, on the same channel. Depending on the mechanisms implemented to fulfil the diverse requirements of the supported traffic flows, the design of TSN networks may require complex and time-consuming configurations [2], [3] through the usage of Satisfiability Modulo Theories (SMT) solvers [4] or mapping tools based on heuristics [5].

This paper focuses on TSN for in-car automotive communications. The motivating scenarios include autonomous driving applications, such as obstacle detection and traffic sign recognition, which generate event-driven traffic that requires bounded and very low end-to-end delays. Such delays range from hundreds of microseconds to tens of milliseconds [6] but, as of today, the TSN support to different classes of event-driven real-time flows is limited. In addition, the need to meet the timing requirements of real-time traffic flows with different generation patterns, i.e., periodic, sporadic and event-driven, suggests to explore the case for online scheduling approaches over TSN as an alternative to complex offline configurations of the network parameters. Moreover, offline scheduling requires that all the characteristics of the flows are known in advance, but in automotive applications not all the flows start at the system startup or at known points of time. New flows can be actually activated during the system operation by an event whose occurrence is not known a priori, e.g., when trailers are connected to cars or trucks. For example, in the work [7], when the trailer is coupled to the vehicle, the Advanced Driver Assistance Systems (ADAS) need to handle the new real-time flows generated by the rear-view camera and park distance control sensors installed on the trailer. To support applications like the ones described above, in-car networks may benefit from online scheduling algorithms, as such algorithms take the decision at run time based on the actual frames to be transmitted. In particular, if the scheduling decision is based only on the frame absolute deadline, which depends on the frame arrival time and is known at run time, all the real-time traffic classes, i.e., periodic or event-driven, can be managed in a uniform way.

The authors are with the Department of Electrical, Electronic and Computer Engineering, University of Catania, 95125 Catania, Italy. e-mail: {gaetano.patti, lobello, luca.leonardi}@unict.it

The novel contribution of this paper is the introduction of online Earliest-Deadline First (EDF) scheduling in TSN to support real-time event-driven traffic. Based on our knowledge, this topic has not been addressed in previous work so far.

The presented approach, called Deadline-TSN (D-TSN), offers the following advantages: (i) no need for offline schedule calculations, as it is an online scheduling approach, (ii) the support for multiple classes of event-driven real-time traffic, thanks to the availability of multiple priority levels, and (iii) no additional frame overhead, as the standard Ethernet frame format is maintained. This paper presents and discusses

- The design of Deadline-TSN.
- The relevant worst-case response time analysis.
- Simulative assessments in realistic automotive scenarios.

The paper is organized as follows. Section II deals with related works, while Section III provides a background on the TSN standards on which Deadline-TSN builds upon. Section IV presents the Deadline-TSN design. Section V provides the worst-case response time analysis,while Section VI presents the results of simulative assessments obtained through OMNeT++ simulations. Finally, Section VII gives conclusions and hints for future works.

II. Related Work

Several works in the literature addressed the suitability of the TSN standards for industrial and automotive contexts [8]–[10]. In particular, as in these scenarios the timing constraints of the real-time traffic flows have to be guaranteed, a number of research works focused on schedulability analysis and schedule generation algorithms for traffic flows in TSN-based networks [11]–[15]. Some works [16]–[18] proposed offline EDF scheduling of periodic traffic flows over TSN. These works exploit the Time-Aware Shaping (TAS) under the assumption that the arrival times and periods of the frames are a-priori known. Consequently, such approaches cannot deal with event-driven real-time traffic, as the frame arrival times of event-driven flows are not known a priori. Conversely, Deadline-TSN supports both periodic and event-driven real-time traffic in a uniform way, as it schedules the frames online based on their arrival times and absolute deadlines.

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3184069

Several approaches were proposed for scheduling periodic real-time flows over TSN networks. For example, the algorithms presented in the works [19] and [20] deal with scheduling of time-triggered flows for industrial communications in the context of Time-Sensitive Software-Defined Networks. Moreover, the bandwidth partitioning system proposed in [21] enables to introduce new flows in TSN-based in-vehicle networks without affecting the already existing traffic. However, unlike Deadline-TSN, none of the above mentioned approaches addresses event-driven real-time traffic flows. In [22] a framework that automates offline scheduling in TSN networks through an SMT-based solver is presented. However, such a framework cannot cope with event-driven traffic and it also entails complex and time-consuming schedule calculations. As far as providing support to real-time event-driven traffic is concerned, the work in [23] proposes a traffic management scheme, called EDSched, which introduces explicit support for event-driven (ED) real-time traffic. EDSched guarantees temporal isolation to time-driven flows using the transmission gates of the IEEE 802.1Q-2018 standard and introduces a novel traffic class, the ED class, to reduce the delay of event-driven flows. However, unlike Deadline-TSN, EDSched is not able to distinguish between the diverse classes of event-driven flows. In [24] a joint algorithm that fragments messages into frames of optimized sizes and schedules them under hard real-time constraints is presented. The approach combines the fragmentation of messages with no-wait scheduling to reduce the latency. In small networks it uses Optimization Modulo Theories solvers to find the optimal configuration in the source nodes, while in case of large networks it adopts an iterative algorithm. Conversely, Deadline-TSN does not require complex calculations in the end-nodes and in the switches, as the frames are scheduled according to an EDF-based policy. A different approach to support real-time traffic without the need for clock synchronization is the Asynchronous Traffic Shaping (ATS), introduced by the IEEE 802.1Qcr-2020 [25] standard. The ATS applies a token-bucket shaping on a per-flow basis and can also be adopted for the transmission of event-driven real-time traffic. According to the ATS specifications, each flow is associated to a shaper instance and has its own bucket size (which defines the maximum burst size that can be transmitted) and token generation rate (which determines the bandwidth assigned to the flow). The ATS assigns to each shaper an eligibility time, after which the head frame of the shaper's queue is sent to the transmission queues. The work in [4] proposes a design-time solution to configure the ATS parameters, such as the assignment of the real-time flows to the queues and the specific priority level to each queue, which exploits an SMT solver. However, as the SMT solution does not scale well for large networks, it may take a long time to compute a schedule [26]. Conversely, Deadline-TSN enables transmission scheduling of diverse kinds of real-time traffic flows without resorting to complex calculations and configurations.

III. Background

Deadline-TSN combines the Per-Stream Filtering and Policing (PSFP) and Strict Priority scheduling mechanisms of the IEEE 802.1Q-2018 standard [27] with the clock synchronization provided by the IEEE 802.1AS-2020 standard. Figure 1 shows a scheme of the forwarding process with the first two mechanisms.

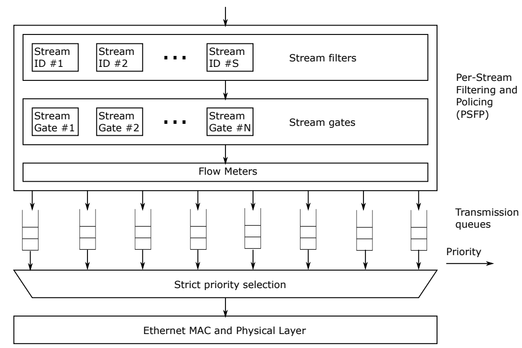

Fig. 1. Forwarding process combining the PSFP and the Strict Priority selection algorithm

According to the IEEE 802.1Q-2018 standard, each Ethernet port maintains up to eight transmission queues, ordered by priority. The PSFP allows for stream filtering, policing and queueing decisions and defines three main components on top of the transmission queues, i.e., the stream filters, the stream gates, and the flow meters. The stream filters associate each frame to a stream and the mapping is made on the basis of a stream_handle value (as defined in the IEEE 802.1CB standard) and/or on the Priority Code Point (PCP) value encoded within the VLAN tag of the processed Ethernet frame. Each stream is, in turn, mapped onto a single stream gate. The flow classification rules may use one or more fields, such as the Destination MAC address, Source MAC address, Virtual LAN Identifier (VID), and Priority [27], to associate a stream with a single stream gate. Stream gates maintain a state value (i.e., open or closed) and an Internal Priority Value (IPV). If the stream gate state is open, the frames mapped onto the stream gate are allowed to be enqueued in the transmission queue corresponding to the current IPV of the stream gate. Conversely, if the stream gate is closed, the frames mapped onto the stream gate are dropped. The stream gate state values and the IPVs can change at runtime, according to a predefined time schedule that cyclically repeats. The stream filters also map a stream to a flow meter, which is in charge of marking the frames, on the basis of defined stream specifications, and dropping them, if the frames do not comply with these specifications. Flow meters are not detailed here, as they are not needed in Deadline-TSN. When a frame arrives at the ingress port of the switch, the stream filters map the frame to a stream gate. Stream gates cyclically execute a stream gate control list (there is one list for each stream gate). Each entry of the gate control list specifies the stream gate state and the IPV, which is used to map the frame to the transmission queue with the corresponding priority. For example, if the IPV of a stream gate is 7, the incoming frames traversing the stream gate will be inserted in the highest priority transmission queue. At the end of PSFP process, the frames are enqueued in the transmission queues according to the IPV of the traversed stream gate. The Strict Priority selection algorithm picks for transmission the frame that is the head of the highest non-

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3184069

empty priority queue and transmits it.

The following Section describes how Deadline-TSN makes use of the above described mechanisms to realize EDF scheduling.

IV. Deadline-TSN

Deadline-TSN inserts the frames in the transmission queues of the Ethernet port based on their absolute deadline, leveraging on the PSFP ability to change the flows' priority following a time schedule. More details are provided in the following.

A. System model and notation

The system model consists of a network made up of switches and end-nodes. The switches are full-duplex, provide multiple Ethernet ports and are fully compliant with the IEEE 802.1Q-2018 and 802.1AS-2020 standards. The end-nodes, instead, are equipped with a single Ethernet port. Both the switches and end-nodes are connected through a full-duplex Ethernet physical layer operating at a fixed data rate (δ). Table I summarizes the notation. Each connection between two nodes is represented by two unidirectional links L<s, r>. Each source node generates one or multiple traffic flows. Each flow F_i is characterized by a period P_i, a relative deadline D_i (i.e., the maximum allowed time-span between the frame generation and delivery to destination), and a source-to-destination path ζ_i, defined as a sequence of H_i links $\zeta_i = \{L_0, L_1, ...L_{H-1}\}$. Each Ethernet port on the path provides multiple transmission queues ordered by priority. Hereinafter, we assume $Q = 8$, i.e., the maximum number according the IEEE 802.1Q-2018 standard, but a lower number of queues does not invalidate the proposed approach. The switches need to be fully compliant with the PSFP, while the end-nodes do not. Deadline-TSN adopts a number N of stream gates to enable deadline-based frame priority. The N value depends on the adopted hardware. In commercial-off-the-shelf switches, the maximum number of stream gates is limited, (e.g., up to 128 in [28]). In Deadline-TSN, N is configured to be a multiple of Q. This way, N/Q stream gates are mapped onto a transmission queue.

TABLE I
Summary of notation

Symbol	Description
δ	Data rate.
L<s,r>	Unidirectional link between the sender node and the receiver node.
F_i	The i-th flow.
P_i	F_i period.
D_i	F_i relative deadline.
H_i	Number of links traversed by F_i from the source to the destination.
ζ_i	F_i path from the source to the destination.
Q	Number of transmission queues in each Ethernet port used in D-TSN.
N	Number of stream gates in each switch used in D-TSN.
VID_i	Virtual LAN Identifier used to map the frame to the i-th stream gate.
G	The index of a stream gate used as parameter in Eq. (1).
$IPV(t, G)$	Internal Priority Value of the stream gate G at time t.
u	Time unit, a constant configurable interval at which the IPV of the stream gates is recalculated.
$f_{i,j}$	The j-th frame of F_i.
$d_{i,j}$	Absolute deadline of $f_{i,j}$.
$a_{i,j}$	Arrival time of $f_{i,j}$.
T_C	Cycle time, defined as $T_C = N \times u$.
$\text{PCP}_{i,j}(t)$	Value used by the sender nodes to assign a frame to a specific queue.
τ_b	Time required to transmit one bit.
$V_{i,j}$	VID value assigned to $f_{i,j}$ to be set in the frame VLAN tag.
R_i	Worst-case response time of F_i.
W_i	Waiting time of F_i at the source to meet Condition (4).
T_{sw}	Switch fabric delay.
C_i	Transmission time of the largest Ethernet frame of F_i.
TQi^{L_y}	Maximum queuing time of a frame belonging to F_i in the transmission port of the y-th link.
$Conf\_d_i(t)$	Function used to calculate the blocking and the interference of a frame with absolute deadline equal to t.

B. Design

Deadline-TSN provides two frame transmission mechanisms, one for the end-nodes and one for the switches. Such mechanisms differ for the frame parameter that drives the selection of the queue the frame has to be inserted in, as the end-nodes use the PCP value, whereas the switches use the VID value. The source nodes use a software transmission mechanism, which computes the PCP and VID values for each frame and transmits the frame to the Ethernet port within a calculated time window. The VID is encoded in each frame by the source node once and for all. The switches do not play any role in assigning the frame VIDs and shall not modify them. In the Ethernet port of the source nodes, the frames are selected for transmission according to the Strict Priority selection defined in [27]. The switches use a specific configuration of the PSFP that allows to change hop-by-hop, based on the frame VID, the queue in which the frame is inserted. In particular, in the switches the control list of each stream gate is configured so that the frame priority increases as the deadline approaches. This is accomplished by changing the IPVs in each stream gate control list accordingly. The switch configuration required by Deadline-TSN consists in setting the stream gate control lists, i.e., the way the IPVs change over time. This means setting for each stream gate, identified by a VID value, a sequence of {State, Interval, IPV} triplets, which cyclically repeats. Such a configuration is made only once, at the network deployment time, and shall be the same for all the network switches. The PSFP configuration through the IPV associated to a frame determines the priority of the queue in which the frame will be inserted. Consequently, the frame scheduling is defined once the frame is assigned the IPV (i.e., the queue) and inserted in the queue. The scheduling then depends on the Strict Priority selection and on the frame position in the queue, i.e., on the frame arrival time. In fact, the queue will be selected for transmission according to the Strict Priority selection and, within the queue, the enqueued frames will be handled in First-In First-Out (FIFO) mode.

The PSFP in the switches is implemented in hardware.

The details are presented in the following subsections.

1) Switch forwarding and transmission: In Deadline-TSN, a frame arriving to an Ethernet port is assigned to a stream gate based on its VID value, which is a function of the frame absolute deadline (d). Each stream gate maintains an Internal Priority Value (IPV), which specifies the transmission queue to be assigned to the incoming frames. The IPV changes at fixed time instants within the gate cycle, following a stream gate control list that cyclically repeats. This allows for dynamically changing the association between the stream gates and the transmission queues. Deadline-TSN exploits this property to implement EDF scheduling within the switches. This way, the highest priority is assigned to the frames with the closest absolute deadline. Deadline-TSN requires to configure the stream gate control lists only once, i.e., when the network

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3184069

	Q0		Q1		Q2		Q3		Q4		Q5		Q6		Q7	
t=0	1	2	3	4	**5**	6	7	8	9	10	11	12	13	14	15	16
t=u	16	1	2	3	4	**5**	6	7	8	9	10	11	12	13	14	15
t=2 x u	15	16	1	2	3	4	**5**	6	7	8	9	10	11	12	13	14
t=3 x u	14	15	16	1	2	3	4	**5**	6	7	8	9	10	11	12	13
	...		...		...		...		...		...		...		...	
t=15 x u	2	3	4	**5**	6	7	8	9	10	11	12	13	14	15	16	1
t=16 x u	1	2	3	4	**5**	6	7	8	9	10	11	12	13	14	15	16

time (t)

Fig. 2. Deadline-TSN stream gate mapping onto transmission queues over time, with $VID_0 = 1$, $N = 16$ and $Q = 8$. Each table element is a stream gate, its column position shows the queue mapping at a given time, while each row shows how the stream gate mapping changes over time.

is deployed.

In Deadline-TSN, the frames are identified only based on the VID, and each VID is associated to a single stream gate. In Deadline-TSN each switch uses a stream filter to get the frame VID and to associate a frame with a stream gate. For the sake of readability, hereinafter each stream gate is identified by the VID value. The first VID starts from VID_0, which is the VID that is mapped onto the stream gate no. 0. The other streams are identified with consecutive values, i.e., $VID_{N-1} = VID_0 + (N - 1)$. The stream gates are always open, while their IPV is recalculated at a constant configurable interval, here called a time unit (u). Moreover, the time unit is used as the time interval for assigning the priority to a frame based on its absolute deadline. In general, the shorter the time unit the higher the granularity of the absolute deadlines that can be encoded. For example, if a frame has an absolute deadline equal to 50us and the time unit is equal to 10us, such a frame will be assigned the same VID of all the frames with absolute deadlines in the range [41us, 50us]. If multiple frames are enqueued in the same transmission queue, they will be transmitted in FIFO order.

For each VID, the stream gate control list follows a cyclical priority shifting, with a period equal to the number of stream gates used in Deadline-TSN multiplied by the time unit. The number of rules is equal to the number of queues used by Deadline-TSN. Each stream gate G has to be configured so as to change its IPV at time $t = n \times u, \forall n \in \mathbb{N}_0$ according to the following function,

$$IPV(t, G) = \left[\left(\left\lfloor \frac{(s(t) + G - 1 - VID_0) \times Q}{N} \right\rfloor\right) \bmod Q\right] \quad (1)$$

where G is the stream gate index and $s(t)$ is the number of time units elapsed since a reference time t_0 calculated as

$$s(t) = \left\lfloor \frac{t}{u} \right\rfloor \quad (2)$$

The example in Fig. 2 shows a table with the matching between the stream gates and the transmission queues over time (i.e., at every time unit u) with $Q = 8$ and $N = 16$. The VID value of each stream gate is the number inside the table element.

For instance, given the case in Fig. 2 with $u = 100$, at time $t = 0$ a frame with *VID* field equal to 5 is inserted in the transmission queue no. 2, while at time $t = 200$ the same frame will be inserted in the transmission queue no. 3.

The VID calculation, which is up to the end-nodes, is presented below.

2) End-node transmissions: The end-nodes generate frames and assign to them the VID and the PCP values of the Ethernet VLAN tag.

When the j-th frame of the i-th flow (i.e., $f_{i,j}$) is generated, its absolute deadline $d_{i,j}$ is given by the frame arrival time $a_{i,j}$ plus the relative deadline D_i of the i-th flow. The VID value $V_{i,j}$ for $f_{i,j}$ is calculated mapping the absolute deadline to a priority, which is an integer number in the range $\{VID_0, ..., VID_{N-1}\}$, calculated as

$$V_{i,j} = N - \left\lfloor \frac{(d_{i,j} - \tau_b) \bmod T_C}{u} \right\rfloor + VID_0 \quad (3)$$

where τ_b is the time required to transmit one bit. The floor function argument is the length, measured in time units, of the time interval between the absolute deadline and the start of the cycle time, defined as $T_C = N \times u$. T_C represents the scheduling temporal horizon and limits the furthest absolute deadline that will be considered by the scheduler (i.e., that will be assigned a VID) during the current time unit u. The cycle time shifts every u, thus progressively extending the schedule to the frames with further absolute deadlines.

In Deadline-TSN, each sender node is allowed to transmit a frame to the Ethernet port only when the time remaining before the frame deadline expires is lower than or equal to $N * u$ (i.e., T_C) and higher than the time unit duration, i.e., when

$$\begin{cases} d_{i,j} - t > u \\ d_{i,j} - t \leq N * u \end{cases} \quad (4)$$

where t is the current time. Before transmitting a frame to the Ethernet port, the sender node assigns to the frame a PCP value calculated as,

$$\mathrm{PCP}_{i,j}(t) = Q - 1 - \left\lfloor \frac{(d_{i,j} - \tau_b - t) \times Q}{T_C} \right\rfloor \quad (5)$$

where t is the current time, i.e., the instant at which the sender node runs the PCP calculation. It is the PCP in the sender node that determines the transmission queue in which the frame will be inserted, as the PSFP in the end-nodes is not required by Deadline-TSN and, if present, has to be disabled. For example, with $u = 100$, $Q = 8$, and $N = 8$, using Eq. (5) the frame $f_{i,j}$ with $d_{i,j} = 800$ is mapped to (i) the lowest priority queue $Q0$ at time $t = [0, 99]$, as the time remaining before the deadline expires is between 701 and 800, (ii) $Q1$ at time $t = [100, 199]$, as the time remaining before the deadline expires is between 601 and 700, and (iii) $Q6$ at time $t = [600, 699]$, as the time remaining before the deadline expires is between 101 and 200. The frame in the example cannot be transmitted at time $t > 699$, as the inequality (4) is not met.

C. Running example

In the example presented in Fig. 3, two sender nodes (S1 and S2) periodically transmit (with period P_i) three flows (f_1, f_2, f_3) to the receiver node (R) through the switch B. All the links $L_{x,y}$ operate at 1 Gbps, so the transmission time of a maximum-sized Ethernet frame is shorter than $13\mu s$.

The relative deadline D_i of the frames of the i-th flow is equal to the flow period. In this example, the time unit u is set to $10\mu s$, the number of stream gates is eight ($N = 8$),

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3184069

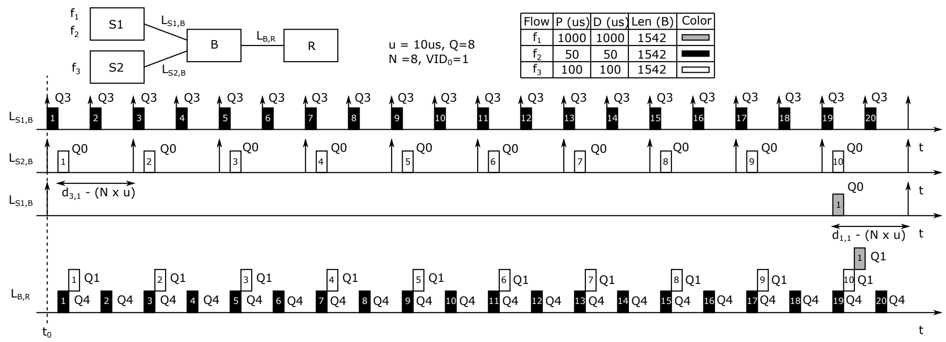

Fig. 3. Running example

and the number of transmission queues used in Deadline-TSN is eight ($Q = 8$). At time $t_0 = 0$ one frame for each flow is generated. The frame $f_{2,1}$ generated by S1 is immediately transmitted to the Ethernet port of S1 because Condition (4) is met, as t_0 is greater than $d_i - (N \times u)$, i.e., $0 \geq 50\mu s - 80\mu s$. The PCP value of $f_{2,1}$ is equal to 3 according to Eq. (5), i.e., $PCP_{2,1} = 8 - 1 - 4 = 3$, so the frame is assigned to the queue no. 3 (Q3 in Fig. 3), and the VID value $V_{2,1}$ is equal to 4 according to Eq. (3). Conversely, before being transmitted to the S2 Ethernet port, the frame $f_{3,1}$ has to wait until the condition $t \geq 100\mu s - 80\mu s$ is met (i.e., $t \geq 20\mu s$) and, according to Eq. (5) and Eq. (3), at time $t = 20\mu s$ its PCP value is set to 0 and its VID value is set to 7. However, at time $t = 20\mu s$, according to Eq. (1), the IPV is equal to 0 for the frames enqueued with $V_{3,1} = 7$, and therefore the frame $f_{3,1}$ will be inserted in the queue no. 0 (i.e., Q0). Finally, at time $t = 920\mu s$ the frame $f_{1,1}$ is transmitted to the S1 Ethernet port, with the PCP value set to 0 and $V_{1,1} = 5$, and mapped to the Q0 queue. In the switch, the frames are inserted in a higher priority queue than the source queue, as their transmission time exceeds the time unit u. As a consequence, during the frame transmissions the stream gates shift their IPV, thus mapping the frame VID value onto a higher priority queue.

D. Discussion on the Deadline-TSN configuration parameters

Deadline-TSN schedules the transmissions based only on the frame absolute deadline. Consequently, it does not require complex network configurations. The only two parameters to be computed are the number of stream gates N to be used, which depends on the adopted hardware and shall be set as higher as possible, and the time unit u, i.e., the interval at which the stream gates calculate (and may change) their IPV.

According to the IEEE 802.1Q standard, the number of queues for each Ethernet port is limited to eight. Consequently, frames with different absolute deadlines may end up in the same queue. A large value for N provides the absolute deadlines encoded in the VID with a finer granularity, which allows to better differentiate between the frames mapped to the same queue. In Deadline-TSN all the frames hop-by-hop shift their priority over time, depending on their VIDs. In particular, the frames with higher VIDs will increase their priority faster than the ones with lower VIDs. As a result, the frames ended up in the same queue at a given hop may eventually split on different queues, based on their VIDs, on the next hops. Consequently, a large value for N is advisable, as it nicely complements the priority shifting mechanism.

The u parameter impacts on both the scheduling granularity, which improves while reducing the u value, and the end-to-end delays, which reduce while increasing the u value. This means that, while increasing u, the number of frames that will be scheduled with the same priority may also increase, if their arrival times are the same. As the frame transmissions have to meet Condition (4), an increase of the u value enables the nodes to transmit earlier, thus resulting in lower delays. In this work, we set the u parameter according to Equation (6), which allows for calculating the time unit (u) so that the first inequality of Condition (4) is always met, otherwise by design the frames would not be transmitted (first argument in Eq. (6)), and at the same time it allows for improving the scheduling granularity, by reducing the time unit until the waiting time at the source node increases (second argument in Eq. (6)).

$$u = \min\left(\min(D_i) - \tau_b, \frac{\max(D_i)}{N}\right) \tag{6}$$

E. Hands-on example

Deadline-TSN is implemented in software in the end-nodes and in hardware exploiting the PSFP in COTS switches. To enable a temporally consistent view of the absolute deadline of each frame among all the end-nodes and switches, they shall be synchronized using the IEEE 802.1AS standard, which guarantees a synchronization error lower than one microsecond. In the end-nodes, when a frame is generated, the absolute deadline is calculated as $d_{i,j} = a_{i,j} + D_i$. If Condition (4) is met, the VID and the frame priority are calculated and assigned to the VLAN Tag of the frame using Eq. (5) and Eq. (3), respectively, and the frame is transmitted to the Ethernet port. Conversely, if Condition (4) is not met, the frame transmission should be postponed to time $t_{tx} >= d_{i,j} - T_C$. At time t_{tx} the VID and the frame priority are calculated and assigned to the VLAN Tag of the frame using Eq. (5) and Eq. (3), respectively, and the frame is transmitted. The switches have to be configured once when the network is deployed using a number N of stream gates. Each stream gate is associated with a VID starting from VID_0 to $VID_0 + N$. The stream gate control list of each stream gate has to be configured so as to switch the IPV according to Eq. (1). For each VID, the stream gate control list follows a cyclic priority shifting and the number of rules is equal to the number of queues (Q) used by Deadline-TSN. For instance, in the running example in Fig. 3, the stream gate control list for VID=1 and VID=4 of the Switch B port that is connected to the node R should be configured as shown in Fig. 4, where *Interval* is the duration of a rule and a State value equal to "o" means that the stream gate is open. No other configurations are required.

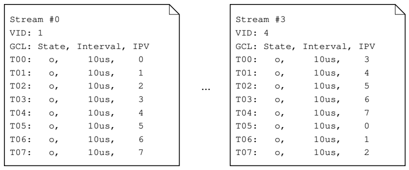

Fig. 4. Stream gates configuration example.

V. Worst-case response time analysis

In this section the Deadline-TSN worst-case response time (WCRT) analysis is presented to verify if a flow set is schedulable. To this aim, the EDF worst-case response time analysis [29], [30] approach is adapted to Deadline-TSN.

In Deadline-TSN a frame is transmitted to the transmission queue immediately after being generated if Condition (4) is met or after a waiting time $W_i = (D_i - T_C)$, if $D_i > T_C$, i.e., if its relative deadline is greater than the cycle time.

$$W_i = \begin{cases} D_i - T_C, & \text{if } D_i > T_C \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The worst-case response time R_i for the i-th flow is defined as the maximum time a frame belonging to the flow takes from its generation at the source node up to the delivery to the destination node. A flow is schedulable if and only if its worst-case response time is lower than or equal to its relative deadline ($R_i \leq D_i$). A flow set is schedulable if and only if all of the flows in the set are schedulable. In Deadline-TSN, R_i is given by the following Equation,

$$R_i = W_i + ((H_i - 1) \times T_{sw}) + (H_i \times C_i) + \sum_{y=0}^{H_i - 1} TQ_i^{L_y} \tag{8}$$

where, as summarized in Table I, T_{sw} is the switch fabric delay, C_i is the worst-case transmission time of an Ethernet frame belonging to F_i, and $TQ_i^{L_y}$ is the maximum queuing time of a frame belonging to F_i in the transmission port of the y-th link. We recall that H_i is the number of links traversed by a frame of F_i from the source to the destination. Here it is assumed that all the frames belonging to F_i follow the same path, i.e., $\zeta_i = \{L_0, ..., L_{H-1}\}$. C_i is calculated as the ratio between the size (in bits) of the largest Ethernet frame of the F_i flow and the network data rate (δ).

$TQ_i^{L_y}$ includes two components. The first one, *blocking*, is due to any lower priority frame being transmitted when the $f_{j,i}$ frame arrives in the transmission port of the L_y link. The second component, *interference*, is due to the frames that are in the higher priority queues or in the same queue when the $f_{j,i}$ frame arrives.

The mapping of a frame $f_{i,j}$ to a transmission queue depends on the frame absolute deadline $d_{i,j}$, which is encoded in the VID field. The frame VID corresponds to a stream gate, which maintains the IPV that, in turn, is mapped on a transmission queue. Due to the limited number of queues in the Ethernet switches, it may happen that frames with different absolute deadlines are mapped onto the same queue. For this reason, to identify the frames that may interfere with the transmission of $f_{i,j}$, a function, called Conf_d(t), is introduced in Eq. (9).

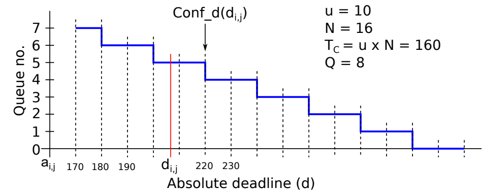

Fig. 5. Graphical representation of the Conf_d(t) function.

$$\text{Conf\_d}(t) = \left\lceil \frac{t \times Q}{T_C} \right\rceil \times \frac{T_C}{Q} \tag{9}$$

Conf_d(t) is used to cap the set of interfering frames for a given frame $f_{i,j}$. In particular, Conf_d(t) calculates the upper bound of a temporal range that starts at the arrival time $a_{i,j}$ of a frame $f_{i,j}$ with a deadline $d_{i,j}$, i.e., $[a_{i,j}, \text{Conf\_d}(t)]$. All the frames with arrival time earlier than $a_{i,j}$ and absolute deadline within the range $[a_{i,j}, \text{Conf\_d}(d_{i,j})]$ are interfering frames for the frame $f_{i,j}$.

Figure 5 shows the graphical representation of Eq. (9). The blue stepped line represents the queue to which a frame $f_{i,j}$ with absolute deadline $d_{i,j}$ is assigned. The figure shows that other frames with absolute deadline higher than $d_{i,j}$, but lower than Conf_d(t), go to the same queue of $f_{i,j}$.

Here the worst-case response time analysis in [29] is applied using the Conf_d$(d_{i,j})$ value to identify the interfering frames for $f_{i,j}$. The worst-case response time analysis adopts the busy period concept and here the busy period of a frame $f_{i,j}$ is defined as the interval $[0, t_e]$ during which all the frames with absolute deadlines lower than or equal to Conf_d$(d_{i,j})$ keep the transmission channel busy, and therefore the frame $f_{i,j}$ is transmitted within t_e. According to Spuri [29], the worst-case response time of F_i is found in a busy period in which all other flows are released synchronously at the beginning of the busy period and then are transmitted at their maximum rate. Hence, we only need to calculate the length of the busy periods of the frames with priority higher than or equal to that of $f_{i,j}$, with different arrival pattern of $f_{i,j}$. The detailed calculation follows the analysis provided in Spuri [29]. Here the Deadline-TSN queuing delay computation is not presented for the sake of space, but readers can refer to the technical report [31].

VI. Simulative Assessment

This Section presents a performance assessment of Deadline-TSN obtained using the OMNeT++ simulation environment, with the NeSTiNg [32] simulator extended to model Deadline-TSN. The performance metrics here considered are the end-to-end delay (E2EDelay) and the deadline miss ratio (DMR). The E2EDelay is defined as the time interval between the frame generation at the source node (*GenTime*) and its complete delivery at the destination node (*RxTime*), measured at the application level. The DMR is defined as the ratio between the number of frames that missed their deadline and the overall number of transmitted frames.

In the following, Deadline-TSN simulative assessments in two different scenarios are presented.

A. Scenario A

The modelled scenario, shown in Fig. 6, is an automotive network with a zonal architecture inspired by the one presented in [33]. It consists of 37 end-nodes, i.e., one Central Controller (CC), four Zone Controller (ZC) ECUs, one Control Unit (CU), four LiDARs (LD), four ultrasonic sensors (US), four radars (RD), eight cameras (CM, MC, RC, DC), one Head Unit (HU), two Rear Seat Entertainment displays (RS), one DVD reader (ME), six speakers (S) and one Telematics module (TLM), connected to six switches. All the links operate at 1 Gbps. The network handles 85 flows and up to 127164 frames per second. Most of the flows consist of Ethernet frame bursts (for example, each Video flow generates one 119-frame burst every 16.6 ms). Three traffic domains are found in this scenario, namely, Multimedia & Infotainment, Control, and Automated Driving & ADAS. Multimedia and Infotainment traffic is transmitted from the ME, HU and TLM nodes to the speakers and RS nodes. This traffic consists of flows relevant to video, audio, and telematics. Conversely, Control traffic is exchanged between the CC and the CU and consists of small frames (i.e., 64-byte long), transmitted with periods ranging from $800\mu s$ to $1s$. Finally, Automated driving and ADAS traffic mainly consists of periodic data samples and Video flows acquired by the sensors (i.e., $RD_{1...4}$, $LD_{1...4}$, $US_{1...4}$, $CM_{1...4}$, DC, RC, $MC_{1...2}$), and event-driven traffic transmitted by the $ZC_{1...4}$.

All the flows from these sensors arrive to the CC, which processes the data and transmits both a Video flow, augmented with additional graphics to assist the driver (DA-CAM), to the HU and Control traffic to the CU.

Moreover, the Video flows from the $MC_{1...2}$ and RC are activated by particular conditions, e.g., during the vehicle maneuvers. The link from SW1 and CC is offered the highest load, i.e., 88% of the bandwidth.

Table II shows the characteristics of the main traffic flows in Scenario A. In Table II, ADAS-Sensors1 and ADAS-Sensors2 are two event-driven flows consisting of frame bursts generated following a random distribution (e.g., maps relevant to obstacle detection), which each ZC transmits to the CC. Video refers to the flows transmitted by all the cameras, while HeartBeat is a critical flow transmitted by all the nodes to report their health status.

In this scenario, Deadline-TSN is compared with Audio Video Bridging (AVB) and with the EDSched approach presented in [8]. In the AVB network, Video and Audio flows

TABLE II
CHARACTERISTICS OF THE MAIN TRAFFIC FLOWS IN SCENARIO A.

Flows	Flow count	Max. Payload (B)	Period (ms)	D (ms)	Type	Workload $L_{SW1,CC}$ (Mbps)
LiDAR	4	1248	1.3	1.3	Periodic	31.4
RADAR	4	625	2.5	2	Periodic	8.9
Ultrasonics	4	188	100	2	Periodic	0.4
Video	8	1500	16.667	16.667	Periodic	702.4
ADAS-Sensors1	4	1500	uniform (10,100)	1	Event-driven	Max 32.9
ADAS-Sensors2	4	1500	uniform (10,100)	1.5	Event-driven	Max 32.9
TelematicsData	1	1500	6	6	Periodic	2.1
HeartBeat	24	64	10	10	Periodic	2.0
DA-CAM	1	1500	0.166	0.166	Periodic	-

are assigned to the highest priority Stream Reservation classes (i.e., SR Class A - queue 7 and SR Class B - queue 6, respectively) and undergo Credit-Based Shaping (CBS). The remaining flows are mapped onto the best-effort classes. Queue 5 is reserved to critical network management traffic, according to the IEEE 802.1Q-2018 standard. Critical flows, such as ADAS-Sensors, HeartBeat, Control, and DA-CAM are assigned to queue 4, while LiDAR, RADAR, Ultrasonics, and TelematicsData flows are mapped onto queue 3. The EDSched network is configured in the same way as AVB, with the difference that the event-driven traffic is mapped onto the ED traffic class (i.e., the highest priority transmission queue) and the bandwidth assigned to the SR traffic is calculated according to the method described in [8].

In this scenario several simulations were run with a different number (N) of stream gates, i.e., $N = 64, 80, 96, 112, 128$. Each simulation was repeated five times with different random generator seeds. The maximum end-to-end delays obtained for the most noteworthy flows are shown in Figures 7, 8, 9, and 10, in which the results are aggregated and the maximum end-to-end delay result is presented for each simulation. The dashed red line represents the flow's relative deadline, while the solid line indicates the WCRT obtained by the analysis.

Fig. 7 shows the maximum E2EDelays obtained by the Video flows. The results show that when the number of stream gates N is equal to 64, the maximum delay obtained by Deadline-TSN is higher than the relative deadline. This result depends on both the number of stream gates (N) and the time unit (u) value chosen in this case, which determine a short cycle time and make it necessary to postpone the transmissions of the generated bursts. Here we recall that T_C limits the furthest absolute deadline that will be considered

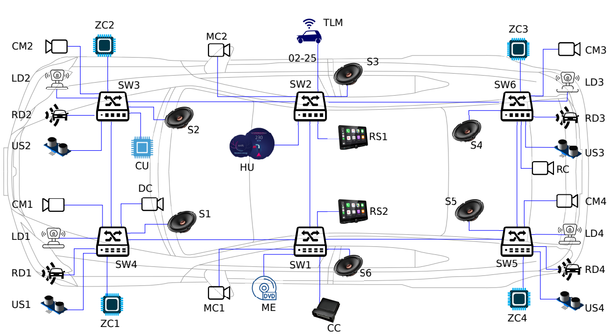

Fig. 6. The simulated scenario.

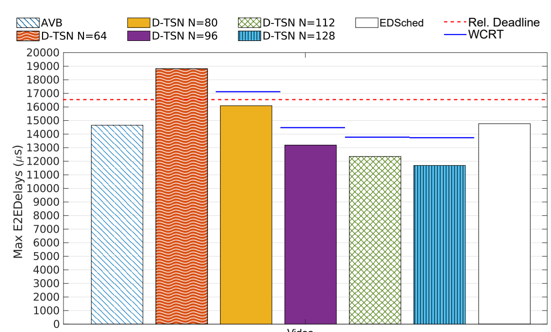

Fig. 7. Maximum end-to-end delays - Video flows

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3184069

by the scheduler during the current time unit u. When N increases, the cycle time also increases, transmissions are not postponed and, consequently, the E2EDelays decrease. In fact, when $N \geq 80$, Deadline-TSN does not experience deadline miss. Moreover, the WCRT values obtained by the analysis show that E2EDelays in Deadline-TSN are lower than the relative deadline with N = 96, 112 and 128. As far as the AVB simulations are concerned, the Video flows always meet the deadline, thanks to both their high priority and the effect of the CBS. This result is not surprising, as AVB was specifically designed to handle this kind of real-time traffic. The EDSched simulations obtained E2EDelays similar to the AVB ones. This is because EDSched adjusts the bandwidth reserved to the Video flows in the SR Class so as to compensate for the higher priority assigned to the ED class. Moreover, both EDSched and AVB take advantage of the CBS when dealing with Video traffic bursts.

Fig. 8 shows the maximum E2EDelays for the HeartBeat, TelematicsData, and Control5ms flows. The simulations results show that all the three approaches, i.e., Deadline-TSN, AVB and EDSched, always meet the deadlines of such flows. In particular, Deadline-TSN experienced no deadline miss with all the N values here considered, whereas the analytical WCRT results are lower than the relative deadline only when $N \geq 96$. In both the AVB and EDSched simulations these flows meet their deadlines. The reason is that, being critical flows, they are mapped onto a high priority best-effort queue, i.e., queue 4 in AVB and 3 in EDSched. Note that the maximum E2EDelay of the Control5ms flow with EDSched is equal to $17\mu s$, but it is not visible in Fig. 8, as it is significantly lower than the other values.

Fig. 9 gives the maximum E2EDelays for the LiDAR, RADAR, and Ultrasonics flows. The results show that with Deadline-TSN the deadlines are always met, while this is not always the case for LiDAR and RADAR flows with AVB and EDSched. This is because such flows are mapped onto queue 3 in AVB and queue 2 in EDSched, and therefore they suffer from the interference of multiple flows, i.e., the critical flows (HeartBeat, TelematicsData, and Control5ms), the Video flows, and the event-driven ADAS flows. Consequently, when an interfering frame burst arrives, such flows experience very high delays (and sometimes exceed the deadlines). Deadline-TSN overcomes such a limitation, as it assigns the priorities online in a frame-by-frame way based on the absolute deadline.

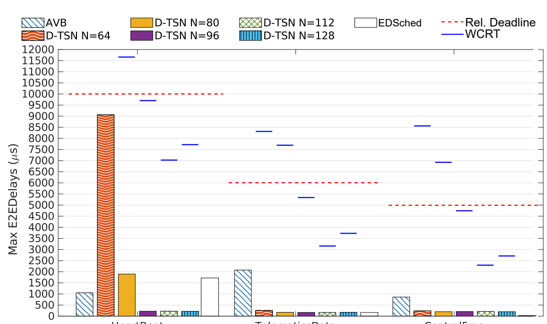

Fig. 8. Maximum end-to-end delays - HeartBeat flows, TelematicsData and Control5ms.

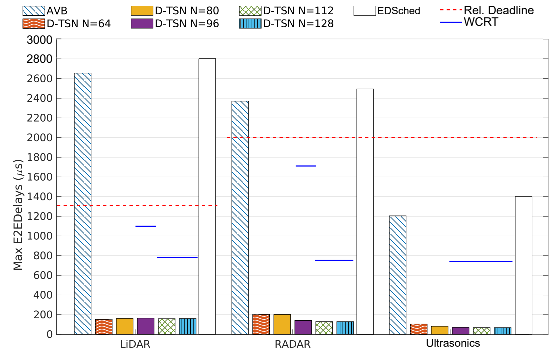

Fig. 9. Maximum end-to-end delays - LiDAR, RADAR and Ultrasonics flows.

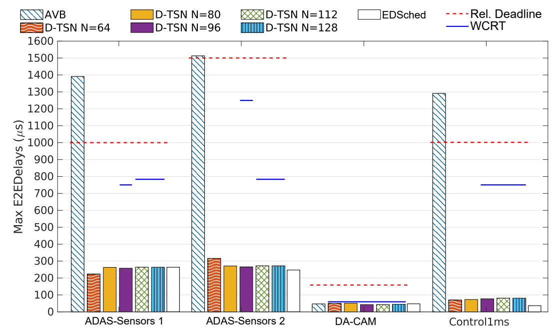

Fig. 10. Maximum end-to-end delays - ADAS event-driven flows, DA-CAM flow and Control1ms flow.

Fig. 10 shows the maximum E2EDelays obtained by the two event-driven flows, i.e., ADAS-Sensors 1 and ADAS-Sensors 2, the DA-CAM flow, and the Control flow with period 1ms (Control1ms). With both Deadline-TSN and EDSched no deadline miss occurs. Moreover, the simulation results of the two approaches are very close to each other, and this means that the Deadline-TSN performance are very good even in comparison with an approach that gives the highest priority to event-driven traffic. Conversely, in the AVB case, although the flows in Fig. 10 are mapped onto the critical flows queue (i.e., queue 4), some deadline miss occurs for the event-driven flows, i.e., ADAS-Sensors 1 and ADAS-Sensors 2, and the Control1ms flow.

Finally, Table III shows the deadline miss ratio obtained via simulation. Note that the flows not reported in Table III do not experience any deadline miss. With AVB, the deadline miss ratio of the ADAS-Sensors 1 and LiDAR flows is higher than 1%, a very high value for such critical flows. With EDSched some deadline miss occurs for LiDAR and RADAR flows. Conversely, with Deadline-TSN some deadline miss occurs for Video flows when N=64, while when the number of stream gates increases, no deadline miss is experienced.

Summarizing, the results of the comparison with AVB and EDSched demonstrate the Deadline-TSN ability to handle real-time flows with very diverse characteristics, arrival patterns,

TABLE III
DEADLINE MISS RATIO RESULTS

	ADAS-Sensor1	**ADAS-Sensor2**	**Video**	**LiDAR**	**RADAR**	**Control1ms**
AVB	1.7%	0.3%	0%	1.0%	0.1%	0.1%
EDSched	0%	0%	0%	0.5%	0.1%	0%
D-TSN (N=64)	0%	0%	1.6%	0%	0%	0%

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3184069

and timing requirements, including event-driven bursts, without the need for traffic shaping.

B. Scenario B

For the sake of comparison with an approach that, unlike AVB, does not bind a flow to a specific traffic class (e.g., Stream Reservation, Best Effort, etc.), but relies on a configuration that takes into account the characteristics of each flow, here a comparative assessment between Deadline-TSN and ATS is presented. The addressed scenario, shown in Fig. 11, is based on the one considered in [34]. In the

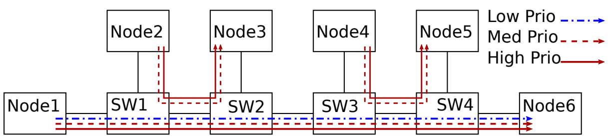

Fig. 11. Network topology and scenario for the comparative assessment.

ATS simulation, the Strict Priority scheduling mechanism is applied on three kinds of flows, i.e., the Low Priority Flows (LP, dashed dotted blue arrow in Fig. 11), Medium Priority Flows (MP, dashed red arrows), and High Priority Flows (HP, solid red arrows). Conversely, in the Deadline-TSN simulation, the frames are scheduled based on their absolute deadline. The data rate is set to 1 Gbps for all the links. In the ATS simulation, the three kinds of flows are mapped onto three different queues of increasing priority (i.e., LP on queue 5, MP on queue 6, HP on queue 7). Transmissions undergo a token-bucket traffic shaping [25] and the shaper parameters of each flow are sized like in [34]. Conversely, in the Deadline-TSN simulation, the relative deadline of each flow is set as shown in Table IV, N is chosen equal to 32 (a realistic value for several switches, e.g., [28]), and the time unit u value is set to $32\mu s$, according to Eq. (6). Note that this is, by design, an unfavorable corner scenario for Deadline-TSN, as the flows are periodic, multiple flows have the same relative deadline, and the arrival time of the first frame is the same for all the flows. The workload on the links that transmit the highest load (i.e., $L_{SW1,SW2}$ and $L_{SW3,SW4}$) is 688.3Mbps. Table IV shows the flow parameters (on the left) and the maximum E2EDelays obtained (on the right).

The results show that the deadlines are met in all cases. Moreover, with Deadline-TSN the E2EDelays for the flow groups [45-47] and [48-54] are 30.7% lower than those obtained with the ATS. In fact, these flows are the LP ones for ATS, and therefore they suffer from the interference of multiple frames belonging to the higher priority flows. Conversely, Deadline-TSN uses a per-frame priority, and therefore the frames of the LP flows will not be delayed by the frames

TABLE IV
CHARACTERISTICS OF THE TRAFFIC FLOWS AND COMPARATIVE SIMULATION RESULTS - SCENARIO B

Flow #	Src. Node	Dst. Node	Payload (B)	Period (μs)	D (μs)	ATS Prio	Max E2EDelay (μs) ATS	D-TSN
[0-4]	N1	N6	719	304	300	HP	216	123
[5-9]	N2	N3	719	304	300	HP	95	93
[10-14]	N4	N5	719	304	300	HP	70	105
[15-24]	N1	N6	1480	609	600	MP	489	429
[25-34]	N2	N3	1480	609	600	MP	338	404
[35-44]	N4	N5	1480	609	600	MP	262	349
[45-47]	N1	N6	80	1000	1000	LP	721	500
[48-54]	N1	N6	1480	1000	1000	LP	736	506

with a higher absolute deadline, thus obtaining lower delays than with ATS. Moreover, while with ATS the frames undergo traffic shaping, in Deadline-TSN there is no shaping, thus improving the bandwidth utilization. Finally, the EDF schedule in Deadline-TSN does not need per-flow configurations, whereas the ATS may require the adoption of SMT solvers [4] to configure on a per-flows basis the maximum token capacity and refill rate of the token bucket and the priority mapping.

To assess the network end-to-end delay performance with different time unit (u) values, the network in Fig. 11 was simulated with $N = 32$ and $u = \{10\mu s, 20\mu s, ..., 100\mu s\}$. Table V shows the obtained results.

TABLE V
MAXIMUM E2EDELAYS WITH DIFFERENT U VALUES

u (μs)	10	20	30	40	50	60	70	80	90	100
Flow #	Maximum E2EDelays (μs)									
[0-4]	458	145	123	123	123	206	244	123	123	160
[5-9]	455	93	93	93	93	168	214	93	93	129
[10-14]	446	111	111	81	91	139	143	81	81	111
[15-24]	783	503	429	429	430	427	446	430	430	430
[25-34]	747	480	404	404	343	404	404	343	343	404
[35-44]	744	477	355	355	318	349	381	317	318	428
[45-47]	1139	835	540	506	501	484	500	427	480	500
[48-54]	1171	866	546	506	506	506	506	506	506	506

When the u value is very low, i.e., $u \leq 20\mu s$, the maximum E2EDelay is high, as the frame waiting time at the source is not equal to zero. When the time unit increases, the E2EDelays decrease up to $u = 40\mu s$. A higher time unit value entails a lower deadline granularity. This means that the same VID, i.e., the same priority, is assigned to frames with quite different absolute deadlines. These frames are scheduled in FIFO order, and therefore their delays also depend on their frame arrival time. Consequently, when u is greater than or equal to $50\mu s$ the maximum E2EDelays show a fluctuating trend. This result is more significant for the flows with shorter relative deadlines, i.e., the flows [0-4], [5-9], and [10-14].

VII. CONCLUSIONS

This work proposed Deadline-TSN, which supports real-time flows with different arrival patterns, including event-driven ones, applying online EDF scheduling without offline schedule calculation, shaping or additional frame overhead. Deadline-TSN is fully compliant with the IEEE 802.1Q-2018 standard and can be implemented on COTS devices with no hardware modifications. Future work will address the integration of scheduled traffic support and experimental evaluations on TSN-compliant switches [28].

ACKNOWLEDGMENT

This research was funded by the Italian Ministry of Research through the PRIN 2017 Program, Project SPHERE - Software architecture for Predictable HEterogeneous REal-time systems, project number 20172NNB4T.

REFERENCES

[1] L. Lo Bello, R. Mariani, S. Mubeen, and S. Saponara, "Recent Advances and Trends in On-Board Embedded and Networked Automotive Systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1038–1051, 2019.

[2] D. Hallmans, M. Ashjaei, and T. Nolte, "Analysis of the TSN Standards for Utilization in Long-life Industrial Distributed Control Systems," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 190–197.

[3] M. Gutiérrez, A. Ademaj, W. Steiner, R. Dobrin, and S. Punnekkat, "Self-configuration of IEEE 802.1 TSN networks," in *22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–8.

[4] J. Specht and S. Samii, "Synthesis of Queue and Priority Assignment for Asynchronous Traffic Shaping in Switched Ethernet," in *IEEE Real-Time Systems Symposium (RTSS)*, 2017, pp. 178–187.

[5] D. B. Mateu, M. Ashjaei, A. V. Papadopoulos, J. Proenza, and T. Nolte, "Letra: Mapping legacy ethernet-based traffic into tsn traffic classes," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2021, pp. 1–8.

[6] M. Alcon, H. Tabani, L. Kosmidis, E. Mezzetti, J. Abella, and F. J. Cazorla, "Timing of Autonomous Driving Software: Problem Analysis and Prospects for Future Solutions," Sidney, Australia, 21-24 April 2020.

[7] M. Haeberle, F. Heimgaertner, H. Loehr, N. Nayak, D. Grewe, S. Schildt, and M. Menth, "Softwarization of Automotive E/E Architectures: A Software-Defined Networking Approach," in *2020 IEEE Vehicular Networking Conference (VNC)*, New York, NY, USA, 18-20 Dec. 2020.

[8] M. Ashjaei, L. Lo Bello, M. Daneshtalab, G. Patti, S. Saponara, and S. Mubeen, "Time-Sensitive Networking in automotive embedded systems: State of the art and research opportunities," *Journal of Systems Architecture*, vol. 117, p. 102137, 2021.

[9] S. Mubeen, M. Ashjaei, and M. Sjödin, "Holistic modeling of time sensitive networking in component-based vehicular embedded systems," in *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2019, pp. 131–139.

[10] L. Lo Bello and W. Steiner, "A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1094 – 1120, June 2019.

[11] D. Thiele and R. Ernst, "Formal worst-case performance analysis of time-sensitive Ethernet with frame preemption," in *IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–9.

[12] M. Ashjaei, G. Patti, M. Behnam, T. Nolte, G. Alderisi, and L. Lo Bello, "Schedulability analysis of Ethernet Audio Video Bridging networks with scheduled traffic support," *Real-Time Systems*, vol. 53, no. 4, pp. 526–577, 2017.

[13] Q. Yu, H. Wan, X. Zhao, Y. Gao, and M. Gu, "Online Scheduling for Dynamic VM Migration in Multicast Time-Sensitive Networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3778–3788, 2020.

[14] L. Lo Bello, M. Ashjaei, G. Patti, and M. Behnam, "Schedulability analysis of Time-Sensitive Networks with scheduled traffic and preemption support," *Journal of Parallel and Distributed Computing*, vol. 144, pp. 153–171, 2020.

[15] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Routing and Scheduling of Time-Triggered Traffic in Time-Sensitive Networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4525–4534, 2020.

[16] M. Vlk, K. Brejchová, Z. Hanzálek, and S. Tang, "Large-scale periodic scheduling in time-sensitive networks," *Computers and Operations Research*, no. 137:105512, pp. 1–16, 2022.

[17] X. Jin, C. Xia, N. Guan, C. Xu, D. Li, Y. Yin, and P. Zeng, "Real-Time Scheduling of Massive Data in Time Sensitive Networks With a Limited Number of Schedule Entries," *IEEE Access*, vol. 8, pp. 6751–6767, 2020.

[18] J. Zhang, Q. Xu, X. Lu, Y. Zhang, and C. Chen, "Coordinated Data Transmission in Time-Sensitive Networking for Mixed Time-Sensitive Applications," in *46th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Singapore, 18-21 Oct. 2020, pp. 3805–3810.

[19] N. G. Nayak, F. Dürr, and K. Rothermel, "Incremental Flow Scheduling and Routing in Time-Sensitive Software-Defined Networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2066–2075, 2018.

[20] Z. Pang, X. Huang, Z. Li, S. Zhang, Y. Xu, H. Wan, and X. Zhao, "Flow Scheduling for Conflict-Free Network Updates in Time-Sensitive Software-Defined Networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 1668–1678, 2021.

[21] L. Leonardi, L. Lo Bello, and G. Patti, "Bandwidth partitioning for Time-Sensitive Networking flows in automotive communications," *IEEE Communications Letters*, pp. 1–1, 2021.

[22] B. Houtan, A. Bergström, M. Ashjaei, M. Daneshtalab, M. Sjödin, and S. Mubeen, "An Automated Configuration Framework for TSN Networks," in *22nd IEEE International Conference on Industrial Technology (ICIT'21)*, March 2021.

[23] L. Lo Bello, G. Patti, and G. Vasta, "Assessments of Real-Time Communications over TSN Automotive Networks," *Electronics*, vol. 10, no. 5:556, pp. 1–12, Feb. 2021.

[24] X. Jin, C. Xia, N. Guan, and P. Zeng, "Joint Algorithm of Message Fragmentation and No-Wait Scheduling for Time-Sensitive Networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 478–490, Feb. 2021.

[25] "IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks - Amendment 34:Asynchronous Traffic Shaping," *IEEE Std 802.1Qcr-2020*, pp. 1–151, 2020.

[26] Q. Li, D. Li, X. Jin, Q. Wang, and P. Zeng, "A simple and efficient time-sensitive networking traffic scheduling method for industrial scenarios," *Electronics*, vol. 9, no. 12, p. 2131, 2020.

[27] *IEEE Std. 802.1Q-2018: IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks*, IEEE, May 2018.

[28] Broadcom Inc., *Broadcom BCM56070 - Switch Programming Guide*, Aug. 2020.

[29] M. Spuri, "Holistic Analysis for Deadline Scheduled Real-Time Distributed Systems," *INRIA Rapport de recherche*, no. 2873, Apr. 1996.

[30] L. George, N. Rivierre, and M. Spuri, "Preemptive and Non-Preemptive Real-Time UniProcessor Scheduling," *INRIA Rapport de recherche*, no. 2966, Sept. 1996.

[31] G. Patti, L. Lo Bello, and L. Leonardi, "Technical report: Deadline-TSN worst-case queuing time calculation." [Online]. Available: https://github.com/lucaleonardi91/Deadline-TSN/blob/main/Technical%20report.pdf

[32] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer, and K. Rothermel, "NeSTiNg: Simulating IEEE Time-sensitive Networking (TSN) in OMNeT++," in *International Conference on Networked Systems (NetSys)*, 2019, pp. 1–8.

[33] J. Klaus-Wagenbrenner, "Zonal EE Architecture: Towards a Fully Automotive Ethernet–Based Vehicle Infrastructure," in *Ethernet & IP @ Automotive Technology Day*, 2019.

[34] A. Grigorjew, F. Metzger, T. Hoßfeld, and J. Specht, "A simulation of asynchronous traffic shapers in switched Ethernet networks," in *IEEE International Conference on Networked Systems (NetSys)*, 2019, pp.1-6.

Gaetano Patti (M'13) received the M.S. and the Ph.D. degrees from the University of Catania, Catania, Italy, in 2013 and 2017, respectively. He is an Assistant Professor with the Department of Electrical, Electronic, and Computer Engineering, University of Catania. Since 2012, he has been working in the field of real-time industrial networks, automotive networks, wireless sensor and actuator networks (WSANs), powerline communications, and networks for mobile robot applications. He coauthored more than 40 papers published in the proceedings of international conferences and in important international journals. He is currently a Member of the IEEE Industrial Electronics Society Technical Committee on Factory Automation (Subcommittee on In-Vehicle Embedded Systems).

Lucia Lo Bello (M'02–SM'09) received the M.S. degree in electronic engineering and the Ph.D. degree in computer engineering from the University of Catania, Italy, in 1994 and 1998, respectively. In 2014, she was a Guest Professor with the University of Malardalen, Västerås, Sweden. From 2000 to 2001, she was a Visiting Researcher with the Seoul National University, Seoul, South Korea. She is currently a tenured Associate Professor with the Department of Electrical, Electronic and Computer Engineering, University of Catania. She is responsible for several international and national projects in the area of real-time embedded systems and networks. She has authored or coauthored about 200 technical papers in the areas of her research interests which include automotive communications, time-sensitive networking, industrial networks, real-time embedded systems, and wireless sensor networks.

Luca Leonardi (M'17) received the M.S degree (summa cum laude) in computer engineering and the Ph.D. degree in systems, energy, computer and telecommunications engineering from the University of Catania, Catania, Italy, in 2016 and 2020, respectively. He is a Postdoctoral Researcher with the Department of Electrical, Electronic, and Computer Engineering, University of Catania. Since 2015, he has been researching in the field of real-time communications, Wireless Sensor Networks (WSNs), Low Power Wide Area Networks (LPWANs), and Industrial Internet of Things (IIoT). He coauthored about more than 20 papers published in the proceedings of international conferences and in important international journals.