

Article

# Assessments of Real-Time Communications over TSN Automotive Networks

Lucia Lo Bello <sup>1,\*</sup>, Gaetano Patti <sup>1</sup> and Giancarlo Vasta <sup>2</sup>

<sup>1</sup> Department of Electrical, Electronics and Computer Engineering, University of Catania, 95125 Catania, Italy; gaetano.patti@unict.it

<sup>2</sup> University of Catania, 95125 Catania, Italy; vastagiancarlo@alice.it

\* Correspondence: lobello@unict.it

† Current address: Viale A. Doria, 6, 95125 Catania, Italy.

‡ These authors contributed equally to this work.

**Abstract:** The IEEE 802.1Q-2018 standard embeds in Ethernet bridges novel features that are very important for automated driving, such as the support for time-driven communications. However, cars move in a world where unpredictable events may occur and determine unforeseen situations. To properly react to such situations, the in-car communication system has to support event-driven transmissions with very low and bounded delays. This work provides the performance evaluation of EDSched, a traffic management scheme for IEEE 802.1Q bridges and end nodes that introduces explicit support for event-driven real-time traffic. EDSched works at the MAC layer and builds upon the mechanisms defined in the IEEE 802.1Q-2018 standard.

**Keywords:** IEEE 802.1; Time-Sensitive Networking (TSN); automotive communications; real-time networks; in-vehicle networks; performance evaluation



check for updates

**Citation:** Lo Bello, L.; Patti, G.; Vasta, G. Assessments of Real-Time Communications over TSN Automotive Networks. *Electronics* **2021**, *10*, 556. <https://doi.org/10.3390/electronics10050556>

Academic Editor: Athanasios Kanatas

Received: 30 January 2021

Accepted: 22 February 2021

Published: 26 February 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Automated driving cars are equipped with several different types of sensors, such as radars, ultrasound, GPS, LiDARs, and video cameras, which contribute to obtain a correct perception of the surrounding environment in various driving conditions. These sensors require a transmission channel able to support high bit rates (e.g., video cameras and LiDARs are the most bandwidth-demanding devices) and high synchronization levels. In addition, safety-critical automotive control applications require deterministic and reliable communications. This means that the transmission/reception times of safety-critical data have to be known and guaranteed. To achieve this kind of determinism, the network nodes need to be synchronized. Automotive Ethernet is considered the solution to overcome the drawbacks of current in-vehicle communication architectures, which are characterized by several small networks connected via gateways, and to move towards a homogeneous in-vehicle network [1,2]. In this context, the Time-Sensitive Networking (TSN) set of standards defined by the Time-Sensitive Networking task group of the IEEE 802.1 working group is a useful tool set that enables reliability, determinism, and time synchronization in automotive communications over Ethernet links. TSN networks consist of IEEE 802.1Q bridges with special features, such as time-synchronization, enhanced scheduling, traffic shaping, path selection and reservation, and fault-tolerance. These features make TSN networks very suitable for supporting deterministic communications in automated driving [3]. This property was proved in several studies, such as the one in [4], that addressed the support provided to time-driven transmissions in the presence of other traffic types, e.g., real-time and best-effort.

However, the external world in which the cars move is not deterministic, and therefore unpredictable events may occur and determine unforeseen situations. For this reason, to enable the autonomous vehicle to properly react to such situations, the communication system has to also support event-driven transmissions with bounded and very low delays.

This work addresses the benefits of a novel traffic management scheme for IEEE 802.1Q bridges that provides explicit support for a new traffic class, the event-driven (ED) real-time class. The approach here presented augments the IEEE 802.1Q-2018 standard but is not part of it. Here ED traffic consists of real-time event-driven flows, whose packets are generated by the occurrence of events, arrive in bursts, and have to be delivered with bounded and low delays. The considered approach, here called EDSched, works at the datalink layer and relies on the forwarding mechanisms defined in the IEEE 802.1Q-2018 [5] standard, with suitable modifications.

The contributions of this work are summarized as follows: (1) the detailed description of the EDSched traffic management scheme, (2) an example of how to apply it to a realistic use case in an automotive scenario, (3) a performance assessment of EDSched through OMNeT++ simulations, and (4) a comparison with the results obtained by an alternative approach in the same scenario.

## 2. Background

According to the IEEE 802.1Q standard, for each port of a switch a maximum of eight FIFO transmission queues, ordered by priority, is foreseen. When the channel is idle, the transmission selection is performed by the Strict Priority algorithm, which selects for transmission the frame that is the head of the highest nonempty priority queue.

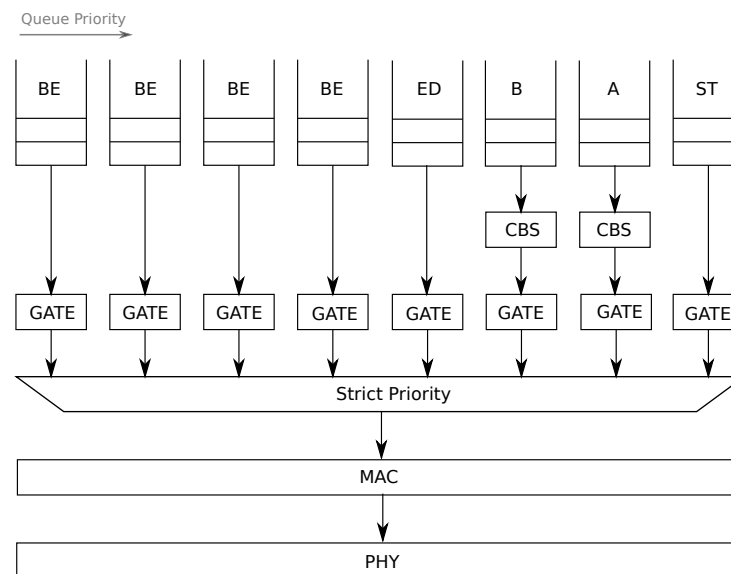
Explicit support for real-time traffic in IEEE 802.1Q switches was introduced by the protocols of the Audio Video Bridging (AVB) standard, where the Stream Reservation (SR) Classes, i.e., classes of periodic flows with reserved bandwidth, have guaranteed bounded low delays. To prevent the SR traffic bursts from exceeding the bandwidth reserved to them, the SR frames undergo traffic shaping, according to the Credit-Based Shaper (CBS) rules. The CBS associates to each SR traffic class a *credit* parameter. The frames that are pending in the SR queues can be transmitted only when their associated credit is higher than or equal to zero. During frame transmission, the credit decreases at the *sendSlope* rate configured for the corresponding SR class. Conversely, the credit increases at the *idleSlope* rate configured for the class when either the credit is lower than zero or the frames of that SR class are waiting for transmission on a busy channel.

The TSN suite of standards, leveraging the common notion of time realized in the network by the IEEE 802.1AS protocol, introduced the support for Scheduled Traffic, i.e., a class of time-driven traffic that exploits a time-aware shaper. Such a shaper is based on the use of *gates*. A gate is associated to each transmission queue of a switch port, as in Figure 1 and maintains a state, either *open* or *closed*. A queued frame can be selected for transmission only if the gate associated to the relevant queue is open and the frame transmission is due to finish before the next gate closing time for that queue. Gates change their state over time, following an ordered Gate Control List, which repeats every *CycleTime*. The gate mechanism provides ST traffic with temporal isolation, i.e., time windows exclusively reserved for transmitting ST frames according to a time schedule, with deterministic delay.

The question tackled in this paper is how to handle event-driven traffic with real-time constraints over IEEE 802.1Q-2018 switches on a network that is shared with other traffic types (ST, SR, and BE). In particular, the goal is to set the conditions that make it possible to guarantee small bounded delay values for ED flows.

Unfortunately, neither AVB nor TSN standards explicitly foresee mechanisms for guaranteeing very low delay to real-time event-driven traffic. In the context of systems based on the IEC 61850 standard, that is relevant to communication protocols for intelligent electronic devices at electrical substations, the model in [6] maps event-driven traffic on the highest priority best-effort (BE) classes to avoid inefficient network bandwidth utilization. The idea is good, but that paper does not deal with SR traffic, as it only covers ST and ED traffic. Note that under suitable assumptions for ED traffic (e.g., a defined minimum interarrival time and a known burst size), if the ED traffic is served in the highest priority best-effort class (as shown in Figure 1), it is possible to calculate the worst case end-to-end delay for the ED traffic by applying response time analysis [7–9]. However, in the presence

of SR traffic, the delays for ED traffic can be large [10]. This is the reason why in EDSched a novel traffic class for real-time ED traffic is introduced and is assigned the second highest priority in the transmission selection algorithm.



**Figure 1.** Alternative approach: event-driven (ED) traffic as best-effort (BE) highest priority.

An interesting property of the TSN family of standards is that it is a flexible toolbox that allows a designer to combine multiple standards to achieve a given performance goal, such as, bounded delay or high reliability [11]. Here, the EDSched mechanism can also leverage the frame preemption functionalities defined in the IEEE 802.3br [12] and IEEE 802.1Qbu [13] standards, respectively, that allow an express frame (high-priority) to interrupt the ongoing transmission of a preemptable frame (lower priority). Preemption does not disrupt the transmission of the preempted frame but simply splits it into fragments.

The EDSched approach here considered can exploit both the support for scheduled traffic and frame preemption. The former provides time-driven flows with temporal isolation, while the latter reduces the delay of high-priority traffic. Their combination in EDSched, together with the introduction of the ED class, makes it possible to guarantee bounded and very low delays to ED traffic. However, as it will be explained in the following Section, EDSched may work with or without preemption.

The idea of reserving a class for ED traffic and handling it accordingly was presented in [14]. However, that document simply lists the steps of the method, without any application to real use cases and without any performance evaluation or comparative assessment. Conversely, this work offers both a full-fledged description of how to apply the EDSched traffic management scheme to real use cases and a performance assessment in a realistic automotive scenario, based on OMNeT++ simulations in multiple configurations. In addition, the work gives the results of a performance comparison, with and without preemption enabled, between the EDSched and an alternative traffic management system that is based on the IEEE 802.1Q standard.

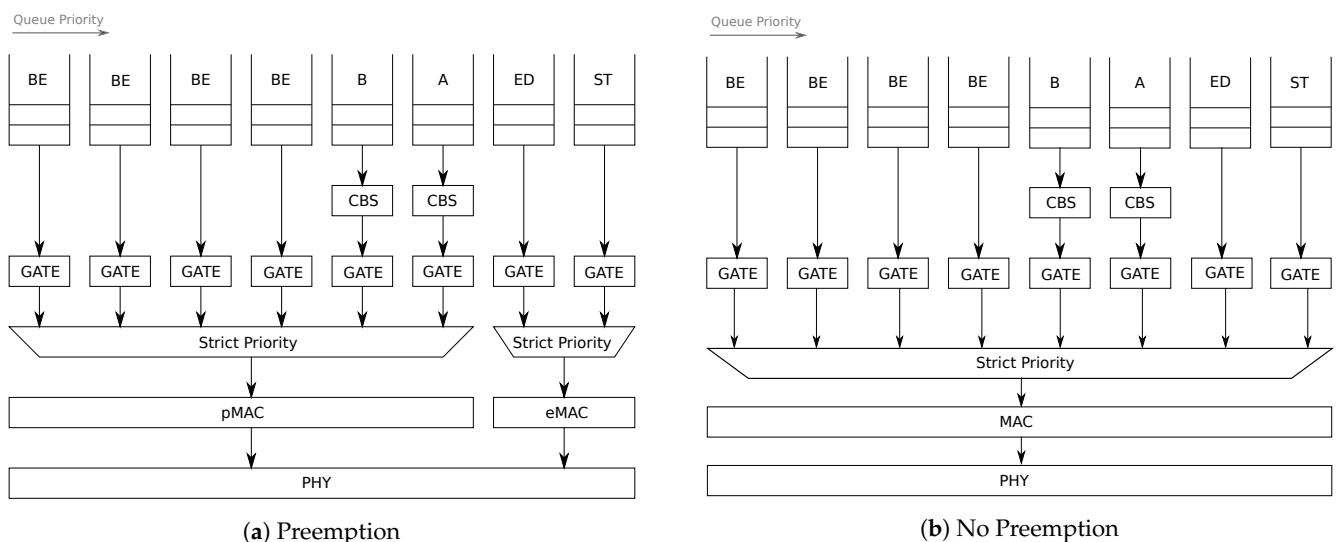
### 3. System Model

The EDSched traffic management scheme here addressed introduces a novel traffic class, the event-driven one (ED). Such a class is assigned the second highest priority according to the Strict Priority transmission selection algorithm that is run at the switch port. The model includes the following traffic classes, which are listed in a decreasing priority order.

- Scheduled traffic (ST): High-priority real-time traffic to be transmitted according to a predefined time schedule (i.e., time-driven), with no interference from other traffic. Both the frame size and arrival times of ST flows are known.
- Event-driven traffic (ED): Aperiodic real-time traffic, including bursts, which is generated by events and requires guaranteed very low delay and low delay variation.
- Stream Reservation (SR): Periodic real-time traffic that requires a guaranteed bounded delay.
- Best-effort traffic (BE): Nontime critical traffic that does not require guarantees on timely or reliable delivery.

Figure 2 shows how the above mentioned traffic classes are mapped onto the eight priority queues of a sample switch port, with (Figure 2a) and without preemption (Figure 2b), i.e., when Ethernet frame preemption, as defined in the IEEE 802.3br and the IEEE 802.1Q-2018 standards, is enabled and when it is not enabled, respectively. In particular, in Figure 2a both the ED and ST transmission queues are linked to the *express MAC* (eMAC), i.e., the MAC level defined in the IEEE 802.3br standard for the express frames, while the queues of all the other classes are linked to the *preemptable MAC* (pMAC), i.e., the MAC level that is defined for the preemptable frames.

In both the models in Figure 2, ST traffic is mapped on the highest priority queue and is temporally isolated from the other traffic classes.



**Figure 2.** ED class mapping on the priority queues of a sample switch port. Case (a)-Preemption enabled. Case (b)-No preemption.

In the following, the mechanisms used by EDSched to enforce temporal isolation for ST traffic in the two models in Figure 2 are explained. Figure 2a shows the combination of scheduled traffic support and preemption.

Here the ST frames need to be isolated from both the preemptable (SR, BE) and nonpreemptable (ED) frames.

To maintain the ST frames temporally isolated from the preemptable frames (SR and BE), the IEEE 802.1Q-2018 standard foresees that the gates associated to the preemptable queues be always open and also defines the *Hold and Release* mechanism. Such a mechanism realizes a fixed interval (here we call it a GuardBand) before the planned transmission of each ST frame such that (i) if there is an ongoing transmission of a preemptable frame, the latter is interrupted according to the standard (e.g., when at least 60 bytes of the frame have already been transmitted and more than 63 bytes remain to be transmitted), (ii) if no preemptable frame is being transmitted, no preemptable frame transmission is allowed to start. Such a GuardBand is sized so as to accommodate the longest nonpreemptable fragment, which is equal to 123 bytes (i.e., 60 + 63 bytes) according to the standard [5].

Conversely, to isolate the ST traffic from the ED frames, which in our model are of the express type (i.e., nonpreemptable), the gate of the ED queue is set to close when the ST gate opens, and vice-versa. This way, if the finishing time of an ED frame transmission exceeds the gate closing time of the ED queue, the ED frame is not transmitted.

In EDSched, preemption entails lower delays for ED frames, as they can preempt the ongoing BE or SR transmissions, thus experiencing a shorter blocking time on a busy channel. On the contrary, preemption increases the maximum delay of the SR and BE frames. For example, the ongoing transmission of a frame of the SR or BE type can be interrupted by the transmission of ED frames one or multiple times. Conversely, without preemption an ongoing SR (or BE) frame transmission cannot be interrupted. The benefit of preemption in the model in Figure 2a is therefore that to transmit an ED frame there is no need for waiting until the ongoing transmission of an entire frame from a lower priority traffic class (i.e., SR or BE) completes.

However, it is known from the literature that preemption is worthwhile when the network operates at lower data rates, e.g., at 100 Mbps. In fact, at lower speeds the transmission time of SR or BE frames may be significant (e.g., the transmission time of a single 1530-byte long frame is 122.4 microseconds). Conversely, at higher speeds (e.g., from 1 Gbps onwards) the transmission times become shorter (e.g., 12.24 microseconds for a single 1530-byte long frame at 1 Gbps), and therefore the delay reduction obtained by preemption becomes negligible (i.e., max 11.25 microseconds for a single 1530-byte long frame at 1 Gbps). For this reason, the EDSched can also work without preemption, as depicted in Figure 2b. In this case, the ED class benefits from being served in the second highest priority queue in the system. The ST frame transmissions are temporally isolated by closing the gates of the non-ST queues when the ST transmissions are planned (i.e., when the ST gate opens) and by opening such gates when the ST frame transmissions finish. The model in Figure 2b, compared to the case with preemption in Figure 2a, reduces the delay for the BE and SR traffic at the expenses of the ED traffic delays. In fact, without preemption, the ongoing SR and BE transmissions cannot be interrupted by the ED frames, and therefore the latter ones suffer from higher delays because of the longer blocking due to the transmissions of the lower priority frames.

Although the end-to-end delay for SR Class A and Class B remains bounded and guaranteed, the introduction of the novel ED traffic class affects the transmission and the configuration of the SR traffic classes. To maintain the delay of the SR traffic as close as possible to the one obtained with the standard CBS configuration, the *idleSlope* calculation for the SR classes in EDSched is modified to take into account the bandwidth used by the ED traffic. For a given port, a novel parameter, called *EDBW*, is defined, which corresponds to the widest portion of band that can be used for transmission of a burst of traffic ED. *EDBW* is calculated as,

$$EDBW = \frac{MaxEDBandwidth}{portTransmitRate} \quad (1)$$

where *MaxEDBandwidth* is an upper limit of band that can be occupied by the real-time event-triggered traffic (expressed in bits per second) that may present within an *OperCycleTime* (*OCT*) and *portTransmitRate* is the port data rate. The new *idleSlope* value of an SR class *N* (with *N* equal to A or B) is calculated as,

$$idleSlope(N) = \frac{operIdleSlope(N) \times OCT}{[GateOpenTime - (EDBW \times OCT)]} \quad (2)$$

where, *operIdleSlope(N)* is the *idleSlope* value calculated according to the Stream Reservation Protocol (SRP) defined in the IEEE 802.1Q standard [5], *OperCycleTime* (*OCT*) is the cycle time of the gate opening/closing operations, and *GateOpenTime* is the duration of the gate open state for the class *N* in the *OperCycleTime*. Equation (2) increases the bandwidth assigned to the SR classes, thus reducing their delays. Here we underline that, although Figure 2a refers to a port featuring eight queues, where four queues are assigned to the BE traffic and two to the SR traffic, this is only an example. If the number of queues per



switch port is lower than eight, it will still be possible to implement the EDSched traffic management scheme as long as there is at least one queue for the SR traffic class. The availability of a higher number of queues (like in Figure 2) allows for a better granularity in servicing the flows belonging to the same class (e.g., the different types of SR or BE flows).

One important property of EDSched is its feasibility, as it can be implemented on devices that support the Stream Reservation Protocol with no special hardware modifications. In the ports of the switch and of the end nodes, the frame priority mapping and the CBS configuration parameters have to be set as in Equation (2) in both cases, i.e., with and without preemption. If preemption is enabled, EDSched also requires the configuration of the queues assigned to the express and preemptable transmissions, respectively.

#### 4. Simulative Assessment

This section reports the results of simulations performed using the OMNeT++ simulation tool and the INET framework, which was modified so as to model the TSN mechanisms specified in the IEEE 802.1Q-2018 standard. The simulations aimed to evaluate the end-to-end performance of EDSched in realistic automotive scenarios under different data rates (i.e., 100 Mbps, 1 Gbps, 10 Gbps).

##### 4.1. Simulation Scenario

The simulation scenarios refer to the realistic case represented in Figure 3.

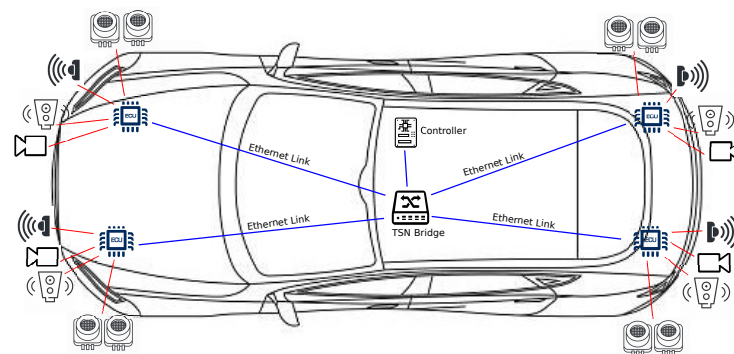


Figure 3. Simulated topology.

where the nodes represent radars, LIDARs, ultrasonics, or cameras. The nodes are directly connected to four peripheral ECUs that transmit raw and aggregated sample data to a centralized controller. Moreover, each ECU transmits to the controller aggregated data, here called ADAS Sensors, in the form of bursts of frames generated according to a random distribution. For instance, such data could be maps relevant to obstacle detection to be fed into the Advanced Driver Assistance System. Each peripheral ECU transmits the same flows as the others. In the assessed scenario, the applications generate messages that, depending on their size, can be transmitted in one or multiple Ethernet frames. Simulation results refer to the end-to-end delay of the entire message, calculated as the difference between the arrival time to destination of the last frame of the message (LastFR<sub>x</sub>) and the message generation time (GenTime), as

$$E2EDelay = LastFR_x - GenTime \quad (3)$$

Here three different scenarios, taken from realistic automotive use cases, are considered:

- **Case A**-100 Mbps: 30 fps compressed video streams, with relative deadline equal to 33 ms.
- **Case B**-1 Gbps: 60 fps compressed video streams, with relative deadline equal to 10 ms.
- **Case C**-10 Gbps: 30 fps uncompressed video streams, with relative deadline equal to 33 ms.

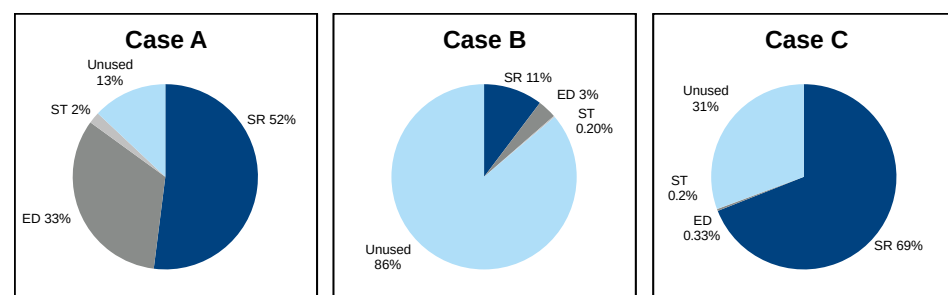
For each flow, Table 1 gives the configuration parameters and the mapping on the traffic classes.

**Table 1.** Flow parameters and mapping on the traffic classes.

Flow	Message Length	Workload	Sampling Time	Arrival Pattern	Class
LiDAR	250 B	0.93 Mbps	10 ms	Periodic	ST
Ultrasonic	100 B	0.23 Mbps	20 ms	Periodic	ST
ADAS Sensors	10 KB	33 Mbps	uniform (10–100) ms	Event-driven	ED/BE *
Video (Case A)	43 KB	52 Mbps	33 ms	Periodic	SR A
Video (Case B)	43 KB	103 Mbps	16 ms	Periodic	SR A
Video (Case C)	6.9 MB	7 Gbps	33 ms	Periodic	SR A

\* ED in EDSched, BE in the alternative approach.

Figure 4 compares the workload and the available bandwidth on the link between the switch and the controller (i.e., the one traversed by the highest load).

**Figure 4.** Workload vs. bandwidth in the three scenarios.

In Case A the overall workload is equal to 87% of the available bandwidth (100 Mbps). The ED traffic workload is equal to 33% of the bandwidth, a very high percentage compared with the other two cases. In fact, this scenario is the one with the highest workload. In Case B the network load is much lower, as the unused bandwidth is equal to 86% of the available bandwidth (1 Gbps). Finally, in Case C the unused bandwidth is equal to 31% of the available bandwidth (10 Gbps), and the ED traffic workload is equal to 0.33% of the available bandwidth, the lowest percentage among the three cases here addressed.

The performance of EDSched are compared with those obtained by an alternative approach that handles the ED flows mapping them onto the highest priority queue in the best-effort class (as it is shown in Figure 1). Both EDSched and the alternative approach are evaluated in the three scenarios here considered, with and without preemption.

#### 4.2. Results and Discussion

The results here reported refer to the cumulative percentage distributions (CPD) of the message end-to-end delays obtained by the ED flows.

The results in Figure 5 (Case A-100 Mbps) show that the maximum delay of ED traffic with the EDSched approach is half the one with the alternative approach. Moreover, EDSched obtained the same results both with and without preemption, i.e., 90% of the messages experienced an end-to-end delay lower than 1.1 ms. Conversely, only about 5% of the messages experienced an end-to-end delay lower than 1.1 ms with the alternative approach.

Preemption does not produce sensible effects on the delay of the ED flows obtained by EDSched, while in the alternative approach preemption slightly improves the maximum delay of ED traffic, from 5640 microseconds without preemption to 5527 microseconds with preemption. This is because the ED traffic is nonpreemptable in EDSched, while it is preemptable in the alternative approach. Consequently, in the latter case the transmission of an ED frame fragment can be accommodated before the gate closing event associated to the expected transmission of ST frames.

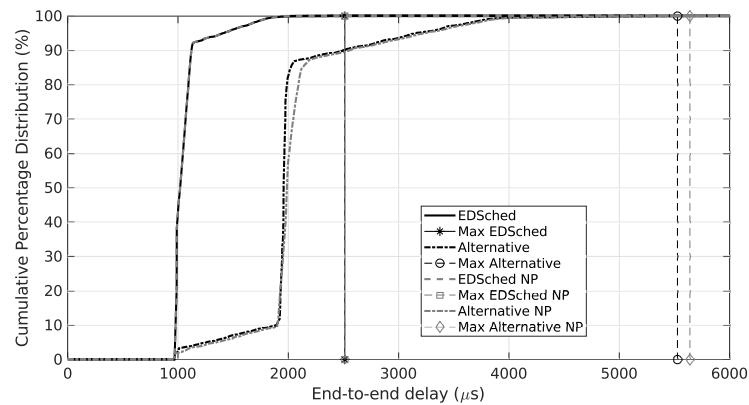


Figure 5. Case A-Event-driven delay cumulative percentage distribution with and without (NP) preemption.

The results in Figure 6 (Case B-1 Gbps) show that the maximum end-to-end delay for the ED traffic with EDSched is lower than 190 microseconds, while with the alternative approach such a value is higher than 210 microseconds. The cumulative end-to-end delay distributions show that with EDSched 90% of the ED messages obtained end-to-end delays lower than 110 microseconds, while only about 50% of the messages experienced similar delays with the alternative approach.

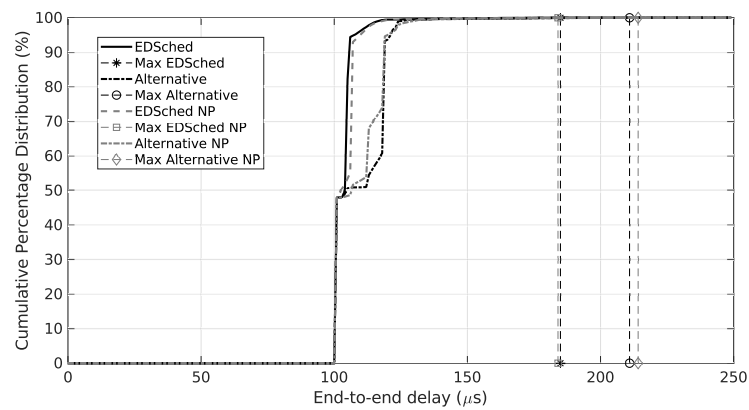


Figure 6. Case B-Event-driven delay cumulative percentage distribution with and without (NP) preemption.

In Figure 7 (Case C-10 Gbps) the end-to-end delay for the ED traffic obtained by the EDSched approach is better than the one experienced with the alternative one. Here the difference between the maximum end-to-end delay values obtained by the two approaches is about 40 microseconds. Considering the 10 Gbps bitrate, such a difference is high, if compared to the very low transmission times.

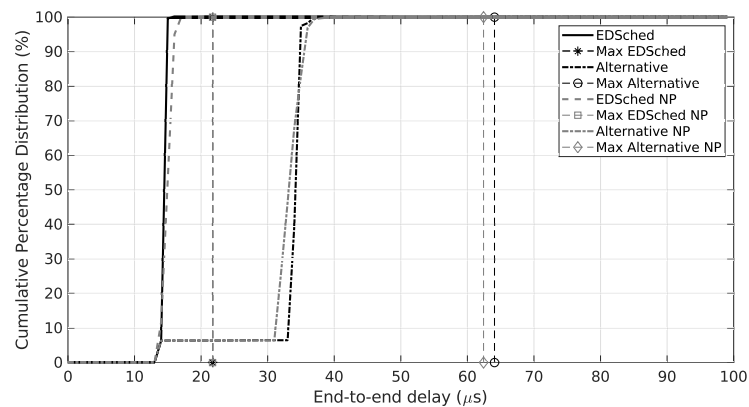


Figure 7. Case C-Event-driven delay cumulative percentage distribution with and without (NP) preemption.



Moreover, in all scenarios here considered the EDSched CPDs show a much steeper increase, thus indicating a lower delay variability than the alternative approach.

Looking at the results in Table 2, the higher the percentage workload-to-bandwidth ratio, the greater the EDSched benefit for the ED traffic (compared to the alternative approach). This is because, while in EDSched the ED traffic does not suffer from the interference due to the SR traffic, in the alternative approach the SR traffic does interfere with ED traffic. This fact determines higher end-to-end delay for the ED traffic and this effect is more significant under high workloads. For this reason, in the cases A and C the percentage improvement achieved by EDSched over the alternative approach is about 55% and 65%, respectively (there is no significant difference between the configurations with or without preemption). Moreover, the greater percentage improvement of EDSched in case C (i.e., 65%) compared to case A (i.e., 55%) depends on the higher SR workload in case C (69% in Figure 4), which implies a higher interference on the ED traffic in the alternative approach than in case A (52% in Figure 4). Conversely, in case B the percentage workload-to-bandwidth ratio is very low, i.e., about 14%. This case is more favorable for the alternative approach, because there is plenty of bandwidth available, and therefore the interference on the ED frames due to the SR traffic is limited. Consequently, in case B the percentage improvement of EDSched over the alternative approach reduces to about 13%.

**Table 2.** EDSched max end-to-end delay improvement for ED traffic.

Case	Workload Bandwidth Ratio (%)	EDSched Max Delay (Microseconds)	Alternative Max Delay (Microseconds)	EDSched Improvement (%)
A (Preempt.)	87	2510	5527	54.59
A (No Preempt.)	87	2512	5640	55.46
B (Preempt.)	14	184	211	12.80
B (No Preempt.)	14	185	214	13.55
C (Preempt.)	69	22	64	65.63
C (No Preempt.)	69	22	62	64.52

Finally, Table 3 shows the maximum end-to-end delays for the other flows, i.e., LiDARs, Ultrasonic and Video flows.

In all the three cases, the results show that the end-to-end delay of ST flows (i.e., LiDARs and Ultrasonic) is not affected by the EDSched transmission mechanism. This was expected, as the ST traffic is served in the highest priority class and is temporally isolated. As far as the video traffic (i.e., SR A flows) is concerned, results show that the deadlines are always met in all the three cases. Moreover, the EDSched impact on the SR traffic, i.e., on the video flows, is limited. In fact, the maximum end-to-end delay obtained by EDSched is only slightly higher than the one obtained by the alternative approach. In particular, the less favorable result (i.e., the one obtained in Case A without preemption) shows that the SR flows experienced a maximum end-to-end delay of 32.9 ms, that is only 1.5 ms higher than the one experienced with the alternative approach (31.4 ms).

In the EDSched approach, in case A the max end-to-end delay of the video messages is lower with preemption than without. This result for the SR traffic depends on the relative balance between the delay increase due to the preemption that the SR frames suffer from the ED traffic and the ability of the SR frames to be split into fragments that can be accommodated in between the ST transmissions, thus reducing the overall SR frame delay. Under low data rates, such as in case A, the benefit exceeds the disadvantage, due to the longer frame transmission times. Conversely, under higher data rates the frame transmission times become shorter, thus reducing the above mentioned positive effect and increasing the negative impact of the preemption from the ED traffic on the SR traffic (as it happens for the Video flows in Case B).

**Table 3.** Max end-to-end delays with (P) and without (NP) preemption for the other traffic classes.

Case A				
Flow	EDSched (Microseconds)		Alternative (Microseconds)	
	P	NP	P	NP
LiDAR (ST)	50	50	50	50
Ultrasonic (ST)	26	26	26	26
Video (SR A)	32,636	32,935	31369	31,414
Case B				
Flow	EDSched (Microseconds)		Alternative (Microseconds)	
	P	NP	P	NP
LiDAR (ST)	10	10	10	10
Ultrasonic (ST)	7	7	7	7
Video (SR A)	8160	8011	7996	8001
Case C				
Flow	EDSched (Microseconds)		Alternative (Microseconds)	
	P	NP	P	NP
LiDAR (ST)	6	6	6	6
Ultrasonic (ST)	5	5	5	5
Video (SR A)	31,010	31,010	31,006	31,006

As far as reliability is concerned, EDSched does not introduce negative effects. In fact, as it is shown in Table 3, no frames are dropped due to deadline miss, as the end-to-end delay is always below the relative deadline. Moreover, no frame loss due to queue overflow occurs. To demonstrate the latter property, the maximum number of frames found in each queue was assessed. Next, we verified if the packet buffer size available in modern switches is sufficient to contain all of them. Table 4 shows the maximum number of frames in each queue on the link that is traversed by the highest load in our scenarios. For the sake of readability, in fact, we reported the maximum number of frames in the queues that belong to the switch port that is linked to the controller. In Table 4 only three out of the eight queues are shown, because in the assessed scenarios there are only three traffic classes, and therefore the other available queues are not used.

**Table 4.** Maximum number of frames enqueued per class at the switch port connected to the controller.

Queue No.	Class	Max. Frames (Case A)	Max. Frames (Case B)	Max. Frames (Case C)
EDSched				
7	ST	1	1	1
6	ED	13	7	6
5	SR A	92	8	15
Alternative Approach				
7	ST	1	1	1
6	SR A	8	4	15
4	ED	17	8	10

We note that there is no large difference in Table 4 between the length of the queues for each class with the EDSched and the alternative approach. The only exception is represented by the SR A queue in Case A (i.e., 100 Mbps), where the maximum number of enqueued frames is equal to 92 with EDSched and 8 with the alternative approach, respectively. The reason for this result is that in EDSched the SR A class is subject to the interference of the ED class. However, we underline that 92 frames is a feasible value for

modern switches. In fact, the maximum Ethernet frame length (i.e., 1530 bytes) multiplied by the maximum number of enqueued frames (i.e., 92) gives a value equal to 140.76 KBytes. Such a memory capacity is widely available in modern switches (e.g., the one described in [15]). This result confirmed that there is no frame loss due to queue overflow.

## 5. Conclusions

The performance assessment of the novel EDSched traffic management scheme showed that ED traffic achieves very low values for both the maximum end-to-end delay and delay variability, with only a very low impact on the delay of the lower priority traffic classes. EDSched can be easily implemented, without hardware modifications, in COTS switches that support the SRP.

Future work will address the EDSched implementation and experimental assessments on the SoC-e switches [16]. We will also investigate the possible integration with a time-aware SDN architecture, such as the one in [17], to set up run-time scheduling configurations based on the flow generation patterns. Finally, as TSN standards integrated with 5G [18] will make several contexts smarter, fully connected and able to meet all key communications requirements, an architecture to integrate our approach with the routing approach for 5G proposed in [19] will be considered.

**Author Contributions:** Methodology, investigation, writing—original draft preparation, and writing—review and editing: L.L.B., G.P., G.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Haeberle, M.; Heimgaertner, F.; Loehr, H.; Nayak, N.; Grewe, D.; Schildt, S.; Menth, M. Softwarization of Automotive E/E Architectures: A Software-Defined Networking Approach. In Proceedings of the 2020 IEEE Vehicular Networking Conference (VNC), New York, NY, USA, 16–18 December 2020; pp. 1–8. [\[CrossRef\]](#)
2. Alderisi, G.; Caltabiano, A.; Vasta, G.; Iannizzotto, G.; Steinbach, T.; Lo Bello, L. Simulative assessments of IEEE 802.1 Ethernet AVB and Time-Triggered Ethernet for Advanced Driver Assistance Systems and in-car infotainment. In Proceedings of the 2012 IEEE Vehicular Networking Conference (VNC), Seoul, Korea, 14–16 November 2012; pp. 187–194. [\[CrossRef\]](#)
3. Mubeen, S.; Ashjaei, M.; Sjödin, M. Holistic Modeling of Time Sensitive Networking in Component-Based Vehicular Embedded Systems. In Proceedings of the 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Kallithea, Greece, 28–30 August 2019; pp. 131–139. [\[CrossRef\]](#)
4. Patti, G.; Lo Bello, L. Performance Assessment of the IEEE 802.1Q in Automotive Applications. In Proceedings of the 2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), Turin, Italy, 2–4 July 2019; pp. 1–6. [\[CrossRef\]](#)
5. IEEE Std. 802.1Q-2018: *IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks*; IEEE: New York, NY, USA, 2018.
6. Docquier, T.; Song, Y.Q.; Chevrier, V.; Pontnau, L.; Ahmed-Nacer, A. IEC 61850 over TSN: Traffic mapping and delay analysis of GOOSE traffic. In Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; pp. 246–253. [\[CrossRef\]](#)
7. Lo Bello, L.; Ashjaei, M.; Patti, G.; Behnam, M. Schedulability analysis of Time-Sensitive Networks with scheduled traffic and preemption support. *J. Parallel Distrib. Comput.* **2020**, *144*, 153–171. [\[CrossRef\]](#)
8. Zhao, L.; Pop, P.; Zheng, Z.; Daigmorte, H.; Boyer, M. Latency Analysis of Multiple Classes of AVB Traffic in TSN with Standard Credit Behavior using Network Calculus. *IEEE Trans. Ind. Electron.* **2020**. doi:10.1109/TIE.2020.3021638. [\[CrossRef\]](#)
9. Ashjaei, M.; Patti, G.; Behnam, M.; Nolte, T.; Alderisi, G.; Lo Bello, L. Schedulability analysis of Ethernet Audio Video Bridging networks with scheduled traffic support. *Real-Time Syst.* **2017**, *53*, 526–577. [\[CrossRef\]](#)
10. Houtan, B.; Ashjaei, M.; Daneshtalab, M.; Sjodin, M.; Mubeen, S. Investigating the Effects of High Priority Traffic on the Best Effort Traffic in TSN Networks. In Proceedings of the 2019 IEEE Real-Time Systems Symposium (RTSS), IEEE, Hong Kong, China, 3–6 December 2019; pp. 556–559. [\[CrossRef\]](#)
11. Lo Bello, L.; Steiner, W. A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems. *Proc. IEEE* **2019**, *107*, 1094–1120. [\[CrossRef\]](#)
12. IEEE Std. 802.3br-2016: *IEEE Standard for Ethernet-Amendment 5: Specification and Management Parameters for Interspersing Express Traffic*; IEEE: New York, NY, USA, 2016.

13. IEEE Std. 802.1Qbu-2016: *IEEE Standard for Local and Metropolitan Area Networks, Bridges and Bridged Networks-Amendment 26: Frame Preemption*; IEEE: New York, NY, USA, 2016.
14. Lo Bello, L.; Patti, G.; Battaglia, F.; Arena, M.; Vasta, G. Method for Managing Traffic in a Network Based Upon ETHERNET Switches, Vehicle, Communication Interface, and Corresponding Computer Program Product. U.S. Patent 10,812,397, 20 October 2020.
15. Broadcom. BCM56072/BCM56071N: Low-Power 440G Switch. 2020. Available online: <https://docs.broadcom.com/docs/56072-56071N-DS1-PUB> (accessed on February 2021).
16. SoC-e. 1G MTSN–Multiport TSN Switch IP Core. Available online: <https://soc-e.com/mtsn-kit-a-comprehensive-multiport-tsn-setup/> (accessed on 26 February 2021).
17. Yang, H.; Zhang, J.; Zhao, Y.; Ji, Y.; Han, J.; Lin, Y.; Qiu, S.; Lee, Y. Experimental demonstration of time-aware software defined networking for OpenFlow-based optical interconnect in intra-datacenter networks. In Proceedings of the 2013 IEEE Globecom Workshops (GC Wkshps), Atlanta, GA, USA, 9–13 December 2013; pp. 440–445. [[CrossRef](#)]
18. Larrañaga, A.; Lucas-Estañ, M.C.; Martinez, I.; Val, I.; Gozalvez, J. Analysis of 5G-TSN Integration to Support Industry 4.0. In Proceedings of the 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; Volume 1, pp. 1111–1114. [[CrossRef](#)]
19. Yang, H.; Liang, Y.; Yuan, J.; Yao, Q.; Yu, A.; Zhang, J. Distributed Blockchain-Based Trusted Multidomain Collaboration for Mobile Edge Computing in 5G and Beyond. *IEEE Trans. Ind. Inform.* **2020**, *16*, 7094–7104. [[CrossRef](#)]