Simulating TSN Traffic Scheduling and Shaping For Future Automotive Ethernet

Zifan Zhou, Juho Lee, Michael Stübert Berger, Sungkwon Park, Ying Yan

Abstract: **The broadening range of applications for vehicles has motivated the evolution of the automotive communication network. Ethernet has been deployed in production vehicles to build in-vehicle networks (IVN) by main manufacturers. To extent Ethernet with real-time service for future E/E architecture, a time-sensitive networking (TSN) profile for automotive Ethernet has been created. This paper evaluates the implementation of multiple traffic scheduling and shaping mechanisms in the automotive Ethernet, respectively. And we especially focus on two solutions, namely the time-aware shaping (TAS) and asynchronous traffic shaping (ATS). To investigate the performance, we introduce a TSN-based automotive gateway testing model in a simulation environment. Furthermore, another two methods, i.e., strict-priority and credit-based shaper (CBS), as well as TAS and ATS are implemented in the model and tested within a domain-based IVN scenario. The results show that TAS guarantees the shortest worst-case latency of high-priority streams, whereas it has a longer transmission latency for low-priority streams. ATS provides less determinism for high-priority streams than TAS, but ensures a better average latency of all streams.**

Index Terms: **Automotive electronics, Ethernet networks, scheduling algorithms, simulation, time-sensitive networking.**

I. INTRODUCTION

A. Background and Motivation

THE evolution of sensor hardware and autonomous control boosts the development of automotive networks. Emerging technologies such as advanced driver-assistance systems (ADAS) and automated driving system (ADS) are widely employed [1]. These automotive applications significantly increase the demand for peripheral devices. In-vehicle components like radar, Lidar, sonar, GPS, odometry, inertial measurement units, and so on are interconnected to perceive the surroundings, as well as to navigate appropriate paths. On the other hand, the computation required by automation is in the range of tera operation per second (TOPS) [1]. Operations generated by real-time applications, such as safety alarm and control messages from the driver assistant system, often have stringent transmission requirements. It is of importance to build a deterministic IVN supporting all types of data communications. Due to high data rate and extensive application, Ethernet is adopted in IVNs to address the demands of interconnecting different components and providing guaranteed services. [2].

Today's IVN is composed of technologies such as controller area network (CAN) [3]/controller area network flexible datarate (CAN FD) [4], FlexRay [7], local interconnect network (LIN) [5] and media oriented systems transport (MOST) [6]. Communication between domains with different protocols is carried out by IVN gateways. Most existing technologies are limited by bandwidth, flexibility, and security [31]. Conceivably, Ethernet as the IVN technology provides speed, an ecosystem for reliable transmission and quality-of-service (QoS), and various features. Fast Ethernet such as 100BASE-T1 standard has been considered as a potential solution, which fulfills the requirements of vehicle networking in terms of costs, cable harness weight, and bandwidth [8]. It is yet infeasible to produce fully-Ethernet architecture. Such transition is taking place with a mixed architecture including Ethernet core and legacy bus systems [32]. However, the best-effort (BE) transmission mechanism of Ethernet makes it unable to ensure deterministic features such as bounded temporal performance, which restricts the capacity of real-time transmission within the networks.

TSN is the technology derived from the IEEE 802.1Q standard providing deterministic transmission over Ethernet. The TSN working group consists of several sections focusing on different aspects. Table 1 lists some examples of the standards. In order to leverage the key features of TSN for IVN, P802.1DG division [10] was created to specify TSN-based profiles that support a wide range of in-vehicle applications, including those requiring security, high availability and reliability, maintainability, and bounded latency. The utilization of traffic shaping and scheduling over TSN guarantees a deterministic low latency and jitter for real-time applications. The IEEE 802.1Qbv standard [11] defines the enhancement for scheduled traffic, so-called time-aware shaper (TAS). It utilizes the time synchronization among all TSN nodes, and reserves transmission resource for critical flows based on their arrival and departure times at every intermediate node. Alternatively, the ATS scheduling approach is an ongoing work outlined in the P802.1Qcr standard [12]. ATS intends to achieve bounded low transmission latency for mixed-type traffics without global time synchronization.

A common assessment method for TSN is building a testing

Manuscript received June 22, 2020; revised October 13, 2020; approved for publication by YU Hua Division III Editor, December 25, 2020.

This research work is partially supported by the project "Time Critical Communication Networks" (no. 8073-00067B) under the International Network Programme granted by The Danish Agency for Science and the Higher Education, and the System Industrial Strategic Technology Development Program (20000307, "Development Over The Air technology that can update ECU software more than 1Mbyte per second") funded By the Ministry of Trade, Industry & Energy(MOTIE), Korea

Z. Zhou, M. S. Berger, and Y. Yan are with the Department of Photonics Engineering, Technical University of Denmark, email: {zifz,msbe,yiya}@fotonik.dtu.dk.

J. Lee and S. Park are with the Convergence Communications Laboratory, Hanyang University, South Korea, email: live_concert@naver.com, sp2996@hanyang.ac.kr.

S. Park and Y. Yan are the corresponding authors.

Digital Object Identifier: 10.1109/JCN.2015.000074

1229-2370/19/$10.00 © 2021 KICS

Table 1. Part of TSN standards included in IEEE 802.1 working group

Reliability	
Frame replication and elimination	802.1CB-2017
Path control and reservation	802.1Qca-2015
Per-stream filtering and policing	802.1Qci-2017
Resource management	
Stream reservation protocol	802.1Qat-2010
YANG data model	802.1Qcp-2018
Link-local registration protocol	P802.1CS
Latency	
Credit based shaper	802.1Qav-2009
Frame preemption	802.3br-2016 & 802.1Qbu-2016
Enhancements for scheduled traffic	802.1Qbv-2015
Cyclic queuing and forwarding	802.1Qch-2017
Asynchronous traffic shaping	P802.1Qcr
Synchronization	
Timing and synchronization for Time-sensitive applications	802.1AS-2020

model and use it in software simulations. It is a flexible solution to build up a network and change the configurations compared with a testbed in reality. The functional blocks in the TSN nodes can be represented by software modules that are programmed to simulate the operations in the actual device. Thus, applying the models in a simulation environment can be used to evaluate the performance in a real system. In this paper, all the scheduling approaches are modeled and evaluated in a simulated IVN environment.

B. Related Work

The research community has reviewed the Automotive Ethernet architecture using different approaches. In [23], the author gives the performance analysis of the Ethernet-based backbone for a future in-car network using the OMNeT++ tool. In [24], the work has explored the timing QoS in the automotive Ethernet network with a focus on the CBS, TAS, and strict-priority using the RTaW-Pegase tool. Another simulation study of adding time-triggered traffics to an Ethernet AVB network is given in [28], upon which the work also gives configuring recommendations to improve IVN performance for vehicle applications. In addition, mathematical analysis is often used to calculate the theoretical latency bound. A formal analysis approach to derive the worst-case timing guarantees of TAS and peristaltic shaper is presented in [25], the work is based on a compositional performance analysis (CPA). Another worst-case latency analysis of TAS using network calculus is proposed in [26]. In paper [27], the authors investigate the performance of TAS, Burst-Control, and Peristaltic shapers for IVN applications using both simulations and analysis. Several studies have also looked into the design of the IVN gateway for automotive Ethernet. In [29], [30], the authors proposed an IVN gateway framework that supports CAN, FlexRay, and Ethernet, as well as a corresponding heterogeneous network synchronization mechanism. The work focuses on the evaluation of the delay deviation caused by the clock drift. The gateway design is used to implement the proposed synchronization mechanism, instead of traffic scheduling and shaping in the gateway. However, none of the above works have studied the performance of the ATS method. In fact, very few researches about the ATS exist since it is yet under the drafting phase.

In [13], the authors propose a performance analysis and comparison between ATS and TAS in an industrial network, also provide a vision on how to improve the TAS mechanism. Instead of an IVN use case, the comparisons are made based on the performance in an industrial network. In [14], a simulation work implementing ATS and frame preemption is presented. The work considers a liner topology consisting of four switches. But it takes ATS, interleaved shaping, strict-priority and preemption into consideration without a comparison with TAS.

Although TSN traffic scheduling and shaping are reviewed from many different aspects in the works mentioned above, a comprehensive comparison among the IVN use cases is still missing, which is considered as one of the primary ATS use cases. Besides, a general testing model of TSN that focuses on scheduler and shaper performance is rarely considered.

C. Contributions

The novelty of this paper is that we propose a testing model to provide an easy approach to evaluate scheduling functions in simulation, then we demonstrate a thorough comparison among the TSN scheduling methods in an automotive Ethernet environment to investigate the impact of using different shaping and scheduling methods in IVN. Specifically, the contributions of this paper are:

- We investigate the usage of TSN gateway with scheduler and shaper in a domain-divided IVN architecture to provide a vision into the automotive Ethernet backbone network and to integrate TSN features into the automotive Ethernet profile.
- We propose a modular testing model of layer-2automotive gateway, which is compatible with various TSN functions, especially the scheduling and shaping methods. Additionally, we implement the strict-priority, CBS, TAS, and ATS modules that can be used in the model and conduct a series of simulations using the testing model with these modules.
- We compare the performance of different combinations of TSN functions. And we show quantitative results of the worst-case end-to-end (E2E) stream latency in a simulated IVN environment. Accordingly, we give our thoughts on the features and limitations of the TSN-enabled gateways in this specific IVN use case.

D. Paper Outline

The rest of the paper is outlined as follow: Section II covers the automotive Ethernet model and discussion about using TSN features. Moreover, we describe the basic principle of the two methods and algorithms used for scheduling. Section III covers an explanation of simulation setup and performance evaluation. Section IV concludes the paper and covers future work.

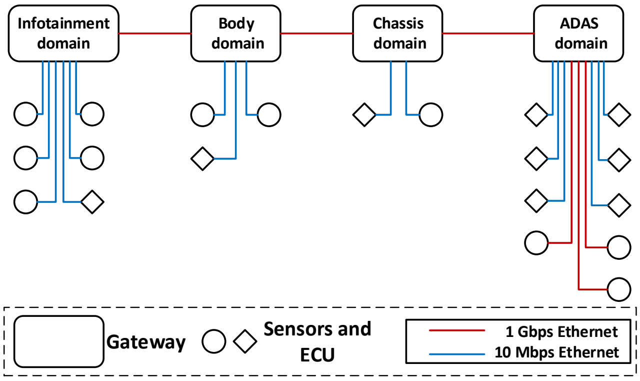


Fig. 1. A domain-based automotive Ethernet framework containing four domains, the sensors and ECUs are connected through the gateways of each domain, all the gateways locate on the backbone of the network. All connections shown in the figure use the Ethernet links.

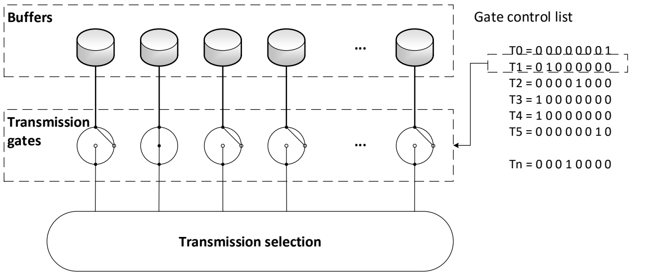


Fig. 2. An illustration of TAS operations: The TAS instance supports up to eight separated buffers, each has an attached gate that controls the outbound traffics. The gates connect or disconnect the buffers from the transmission selection, to allow or prevent frames get selected from the buffers. The status of every gates are defined by the entries in the GCL. The figure shows the status at time T1, when only one gate is open and the others are closed.

II. ANALYSIS OF TRAFFIC SCHEDULING AND SHAPING IN TSN

A. *Automotive Ethernet Framework*

In this paper, we focus on a domain-based IVN topology, as shown in Fig. 1, the network is divided into four sections: Infotainment, body, chassis, and ADAS domain, where each domain consolidates several electronic control units (ECUs) into the domain group. One capable ECU operates as the domain gateway in each group. The gateway connects to all ECUs in the same domain and to another adjacent gateway on the backbone. The backbone provides a connection between domain gateways. Accordingly, communications between ECUs from different domains need to pass through the backbone network.

We assume the IVN under consideration is interconnected by 100 Mbps and 1 Gbps Ethernet links without transforming between protocols. In this topology, gateways operate as a layer-2 bridge, which forwards incoming frames and handles mixed-types of traffics, e.g., time-critical and BE traffics. TSN functionalities, e.g., time synchronization and time-triggered scheduling, are implemented and enabled in the gateway. The arrival traffics in the gateways is classified based on the predefined traffic classes and recognized by using per-stream filtering and policing (PSFP) defined by 802.1Qci [15]. Therefore, the domain-based framework enables the backbone to offer TSN services to different traffics. By modifying the queuing and egress gating mechanisms in the gateway, we implement TAS and ATS shaping for the given IVN framework.

B. *Time-Aware Shaper (TAS)*

In TSN, the time synchronization is based on the IEEE 1588 precision time protocol (PTP). In a PTP domain, the PTP protocol depends on time-stamped frames exchanged between a timing master clock and a timing slave clock, with intermediate boundary and/or transparent clocks to maintain the time accuracy as the 1588 packets traverse the network.

With common time synchronization, transmission actions are executed based on predefined temporal events in time-triggered networks (e.g., time-triggered Ethernet, TSN). Accordingly, a TSN TAS carries out actions when the clock reaches the time instants. For periodic traffic streams, subsequent time-triggered events are placed. The intervals are set to the cycle time of the streams. From the network perspective, TAS is capable of establishing a deterministic data transmission by creating an appropriate schedule based on the knowledge of periodic transmitting events in the system. In this paper, we use a static network topology, where the number of nodes is fixed, and the patterns of streams are consistent. The schedules in TAS are manually configured at the design time.

In TAS, the scheduled traffic (ST) class is a newly introduced traffic class which has the following characteristics: a. highest priority b. scheduled on a per-stream basis in each node c. registration in bridges with operations as well as attribute values for registration and de-registration etc.

Fig. 2 depicts the operations of TAS. As shown, traffics with different classes are splitted and stored into separated queues. Up to 8 queues per port are supported. Each queue has an attached transmission gate that controls the outbound flows. The state of gates is either *open* or *closed* following the entries in gate control list (GCL) within each port. The execution of entries in GCL is performed under the global synchronous timing. As shown in the figure, only one gate is open at time T1 when the other gates are closed. Thus, critical traffics can be assigned with dedicated time slots to get transmitted from the buffers with TAS scheduling.

Interfering traffic remains the main problem that affects the transmission latency because it can occupy an output port relatively long and block the following time-critical traffics. TAS can eliminate the contention by assigning specific time slots for transmission of the ST frames. In this case, the ST frames are ensured with a deterministic low-latency transmission. Guard band needs to be added prior to ST windows so that the influence from interference traffic on ST is limited. The GCL of TAS also supports dynamic runtime configuration, by using a centralized network controller, e.g., software-defined network (SDN) controller [16], [17], the GCLs can be adjusted according to the current network state, more bandwidth is available to other traffic classes, which is an effective way to improve the bandwidth utilization.

Due to the dependence on synchronous global clocks, the performance of TAS is affected by the precision of the synchronization information. If the synchronization does not function appropriately, the TAS gate could miss the transmission window for the ST frame and cause an extra delay. Even worse, the failure is in a chain-like effect where one node will affect the actions in the following nodes. The time for recovering from failure, e.g., switch over to a new master, which possibly causes transmission disorder and frames backlog in the queue. Besides its dependence on the clock synchronization in all nodes, TAS has challenges in mapping the critical application data into the ST class. Automotive applications such as fast diagnostic data used for verifying driving functionality generate samples at a constant rate. The delivery requires a guaranteed deadline. According to the feature of TAS, such cyclic applications with a fixed frame size are able to fit in the ST class. However, applications that generate alarm and event messages are usually sporadic. There might be either a single message or a burst of messages. The applications need low-loss and bounded-delay delivery to one or several end devices. Due to the nature of TAS, it is not feasible to utilize ST class. To address the aforementioned limitations of synchronous scheduling, ATS is designed to achieve real-time performance in TSN with mixed traffic patterns, extending current approaches in real-time Ethernet.

C. Asynchronous Traffic Shaping (ATS)

ATS shapes the traffics in a per-class manner It calculates and maintains the status of every traffic class. ATS relay on an independent clock at each separated node, eliminating the need for synchronization within the network. Based on the shaping algorithm, ATS assigns eligibility time to each traffic class, and a class is eligible for transmission if the assigned eligibility time is less than or equal to the current time. When several classes are eligible to send out frames at the same time, the classes are selected for transmission in ascending order of the eligibility times.

The pseudo-code of the algorithm used to calculate eligibility time is given in Algorithm 1. Deriving from the token based emulation (TBE) algorithm [18], ATS remains the concept of emulating a token bucket for traffic shaping. *Bucket full time* is the time instant when the bucket is full of tokens. The size of the bucket is equivalent to the committed burst size of the traffic class. On the contrary, *bucket empty time* is the time instance when there are no tokens remaining in the bucket. Basically, *empty to full duration* is the time span that the bucket fills up with tokens from empty to full by the committed information rate. *Length recovery duration* denotes the time span that is required to accumulate a number of tokens that is equal to the transmitted frame's length.

Inside a single ATS, the *shaper eligibility time* is the time when the number of tokens in the bucket is more or equal to the *frame size*. Taking into account a group of shapers with the same traffic class, the *group eligibility time* means the most recent *eligibility time* from the previous frame processed by the shaper in the same class. *Max residence time* is used to limit the time a frame residing in one node, a frame is valid only within the *Max residence time*.

As the algorithm indicates, the calculation of the eligibility time of a frame strongly depends on the size of the last transmitted frame and the arrival time of the current frame. Unlike the TBE algorithm, the eligibility time is not only dependent on the number of tokens in the bucket. Instead, the bucket full time and the bucket empty time are considered in the ATS algorithm as well. Thus, ATS can shape the streams based on a timing scale. The ATS algorithm allows a certain scope of bursty transmission. However, it still limits the amount of data during a certain period to avoid accumulating a large frame bulk in the downstream node.

Algorithm 1 ATS algorithm Pseudo code

```
/* Initialization */
    $T_{eligibility} = 0$
    $T_{bucketFull} = 0$
    $T_{groupEligibility} = 0$
    $T_{bucketEmpty} = -(burstSize/\text{rate})$
/* frame Processing */
    $D_{lengthRecover} = frame.length/\text{rate}$
    $D_{emptyToFull} = burstsize/\text{rate}$
    $T_{shaperEligibility} = T_{bucketEmpty} + D_{lengthRecover}$
    $T_{bucketFull} = T_{bucketEmpty} + D_{emptyToFull}$
    $T_{eligibility} = \max(T_{arrival}\ ,$
                    $T_{groupEligibility}\ ,$
                    $T_{shaperEligibility})$
/* Shaping */
if $T_{eligibility} \leq (T_{arrival} + \text{MaxTime}/1.0e9)$ then
    $T_{groupEligibility} = T_{shaperEligibility}$
    $T_{bucketEmpty} = (T_{eligibility} < T_{bucketFull})$ ?
      $T_{shaperEligibility}$ :
    $T_{shaperEligibility} + T_{eligibility} - T_{bucketFull}$ ;
    $AssignAndProcessd(\text{frame}, T_{eligibility})$
else
    $Discard(\text{frame})$;
end if
```

The worst-case delay of the time-sensitive traffic needs to be guaranteed. In the case of TAS, the delay bound is dependent on the length of transmission windows. Whereas for ATS, the delay bound can also be calculated. A formula of queuing delay is given in [12], [18]:

$$D_{BU,max}(k,f) = \max_{h \in F_H(k,f)} \left\{ \frac{l_{\text{MIN}}(h)}{R(k)} + \frac{\sum\limits_{g \in F_H(k,h) \cup F_S(k,h)} b_{\text{MAX}}(k,g) - l_{\text{MIN}}(h) + l_{LP,MAX}(k,h)}{R(k) - \sum\limits_{g \in F_H(k,h)} r_{\text{MAX}}(k,g)} \right\}. \quad (1)$$

In (1), k and f refer to the current hop and the stream of interest. $F_H(k,g), F_S(k,g), F_{LP}(k,g)$ are the higher, same and lower traffic classes of stream f's class. R(k) is the transmission data rate at the kth hop and $r_{\text{MAX}}(k,g)$ is the committed rate of stream g at the kth hop. $l_{\text{MIN}}(h)$ represents the minimal frame length of stream h and $l_{LP,\text{MAX}}(k,h)$ is the maximum size of interference frame of the lower traffic class than

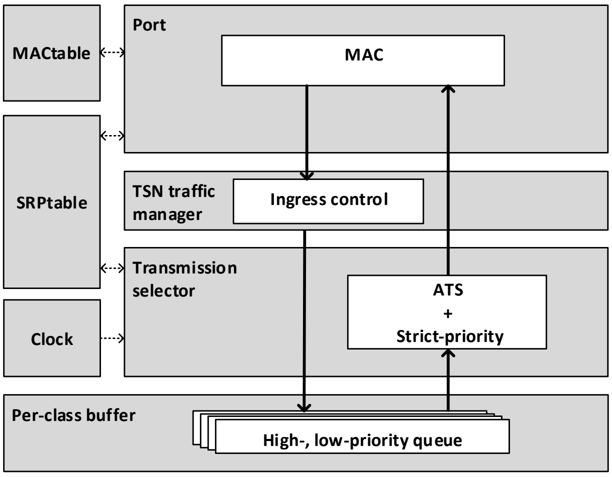


Fig. 3. Functional modules and the connections of the ATS gateway testing model, the receiving process of data flow is represented by the solid lines with downwards arrow, the transmitting process of data flow is represented by the solid lines with upwards arrow, the controlling connections among the modules are represented by the dotted lines.

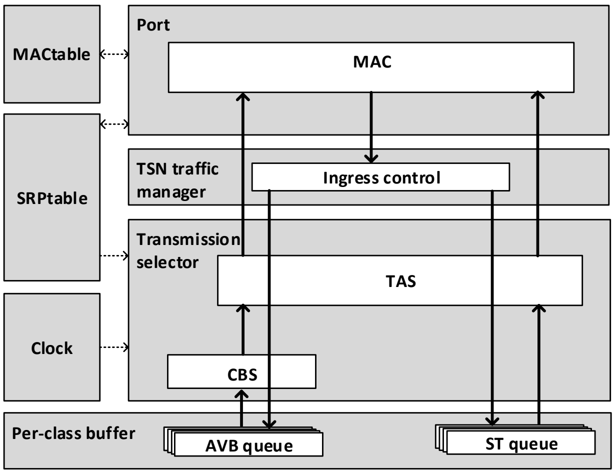


Fig. 4. Functional modules and the connections of the TAS gateway testing model, the receiving process of data flow is represented by the solid lines with downwards arrow, the transmitting process of data flow is represented by the solid lines with upwards arrow, the controlling connections among the modules are represented by the dotted lines.

stream h. $b_{\mathrm{MAX}}(k, g)$ is the committed burst size of stream g at the kth hop. The delay bound considers the worst-case when a maximum-size frame of lower traffic class initiates transmission, plus all higher classes transmit the maximum size of burst after the lower traffic class. In this case, the frame from the stream of interest will be transmitted after the interference frame and all bursts. Based on the formula, it is possible to ensure bounded queuing delay since the burst size of each stream is limited, and using strict priority for transmission selection will enable time-sensitive streams to bypass interference in ATS.

III. Testing Models of TSN-enabled Gateway in IVN

This section presents our design of the models for simulations. We create our own gateway testing models in the OMNeT++ simulator [19] based on the Core4INET framework [20]. We implement and integrate the desired TSN functions for performance analyzing purposes.

The framework of the ATS and TAS gateway testing models are depicted in Figs. 3 and 4, respectively. The models are designed with modularity feature, they are easily customized based on the function requirements. Various TSN features can be added to the model and tested as an IVN gateway function.

A. General-purpose Modules

As shown in the figures, some of the components are compatible in both ATS and TAS models. The functions in these modules are implemented regardless of the scheduling and shaping methods. For instance, the media access control (MAC) module is used in the port. When one frame gets received at the physical port from the network, it is delivered to the MAC module, where the frame is delimited and recognized, as well as a check on the frame check sequence (FCS) is executed. The frame is passed up without modification if the FCS check is correct. At the transmitting process, when a frame is sent out from the queue, it is forwarded to the MAC module. The frame gets transmitted from the MAC when the transmitting port is idle. Otherwise, frames will be queued up in the MAC. The MAC module adds an inter-frame gap between two frames, also fills the empty fields, e.g., source and destination MAC address, in the frame.

Though common modules are utilized in both models, some of them are different in configurations. For example, the clock module in the TAS model counts time and synchronizes with the other nodes in the network. While the synchronization function of the clock module in ATS is disabled, so the clock is only aligned to the local time in the single model. The ingress control module offers classifying functions. It requires the mappings from stream IDs to the supported classes in the models, then forwards the frames to the corresponding queues. The buffering of streams in both models are on a per-class basis. The slicing of buffering space depends on the classification strategy in the model. The main difference in the models is the implementation of transmission selectors and shapers. According to the functional setups, single or multiple selectors and shapers can be added to the model and be applied to different traffic classes. In the ATS model, the shaper module combines both ATS and strict-priority to select frames from queues. On the other hand, two separated modules of CBS and TAS are implemented in the TAS model for transmission selection.

B. ATS Model Description

Receiving process. After the conformance check at the port, the frame is directed to a traffic manager. Followed by the functions defined in the 802.1Qci standard, an ingress control module will make filtering and policing decisions to the incoming frames. In addition, depending on the classification configuration, the control module also reads the stream ID and destination MAC address of the frame and optionally assign an internal priority value (IPV) to the frame. The original priority values and

Table 2. Feature of the simulation streams in the example IVN

Stream no.	Path	Priority	Frame size/interval
1	Info domain → Info domain	Low	11 $B/125\ \mu s$
2	Info domain → Info domain	Low	1250 $B/125\ \mu s$
3	Info domain →ADAS domain	High	60 $B/10\ ms$
4	Info domain →Body domain	Low	2 $B/125\ \mu s$
5	Info domain → Body domain	Low	2 $B/125\ \mu s$
6	Info domain →Info domain	Low	1250 $B/125\ \mu s$
7	Info domain → Info domain	Low	1250 $B/125\ \mu s$
8	Info domain → Info domain	Low	16 $B/125\ \mu s$
9	Body domain → Chassis domain	High	4 $B/10\ ms$
10	Chassis domain → ADAS domain	High	230 $B/10\ ms$
11	ADAS domain → Body domain	Low	2 $B/125\ \mu s$
12	ADAS domain → Body domain	Low	2 $B/125\ \mu s$
13	ADAS domain → Info domain	Low	1250 $B/125\ \mu s$
14	ADAS domain → Chassis domain	High	625 $B/10\ ms$
15	ADAS domain → Chassis domain	High	625 $B/10\ ms$
16	ADAS domain → Info domain	Low	1250 $B/125\ \mu s$
17	ADAS domain → Body domain	Low	1250 $B/125\ \mu s$

IPVs are mapped to the traffic classes supported in the gateway. Later based on the assigned classes, the frames are queued on a per-class basis in the potential transmission ports.

Transmitting process. When a frame is inserted in the egress queue, the ATS will firstly stamp the current time as the arrival time of the frame. Combined with the reserved rate of the corresponding stream, which is stored in the stream reservation protocol (SRP) table [21], ATS calculates the eligibility time for the frame and sets a timestamp on the frame. The calculation complies with the process given in Algorithm 1. Accordingly, a clock module sets a timer of the eligibility time. The clock generates a signal to the shaper when the time is reached. The ATS then selects the frame for transmission and prioritizes the high-priority frame if both classes have eligible frames. When an eligibility time is assigned to a frame, the shaper also updates the status of the corresponding stream, i.e., *bucket full time*, *shaper eligibility time*, etc, and prepares for the next frame.

C. TAS Model Description

Receiving process. After the frames pass through the MAC module in the TAS model, they are forwarded to the ingress control module. The module classifies the frames to ST and AVB classes according to their stream ID. Afterward, separated queues receive the corresponding frames at the potential transmission ports.

Transmitting process. At the transmitting process, ST class operations are defined by the transmitting gate parameters, namely the GCL, in the TAS module. It includes the opening time, opening period, and cycle time of the ST gate operations. By following these predefined operations, the TAS module will forward the ST streams to the port on a timing basis. The CBS module is responsible for regulating the AVB streams from the queues and for calculating the credit of the streams. The SRP table and clock provide the reserved rate of the AVB streams and timing to the CBS module for credit calculation. The CBS can effectively prevent a bulk of consecutive transmission of AVB frames. A description of the CBS working process can be found in [33]. Besides the CBS regulates the AVB streams, the transmitting gates connecting the queues and the ports are controlled by the TAS. The gates for AVB queues are open outside the transmitting window of ST streams.

IV. PERFORMANCE EVALUATION

This section presents our evaluation of TSN scheduling and shaping for automotive Ethernet. We implement the IVN topology given in Fig. 1 within the OMNeT++ simulator. In the simulations, we use our testing models with different configurations to analyze the performance of TAS and ATS in the IVN environment. The device used for running simulations is based on an Intel 2-core i5–6200U CPU @2.30Ghz.

A. Simulation Topology

As mentioned in Section II, the automotive Ethernet built in this paper follows a domain-based architecture. Ethernet connections are used for both the backbone network and the interconnection within domains. 100 Mbps and 1 Gbps full-duplex Ethernet links are used in the simulations to ensure sufficient bandwidth. With the focus on the link-layer protocols, we build most of the network functions up to layer 2 of the OSI model. TSN bridging is deployed at each gateway. TSN traffic scheduler and shaper are deployed at the egress port of the gateways and endpoints (e.g., ECUs and sensors). OMNeT++ is a pure software simulator, therefore, all nodes within the network are synced up with the simulator kernel timing and use it to correct the local clock.

B. Stream Simulation

The endpoints in the simulation are TSN-capable transceivers, which contain applications for traffic sources and sinks. To simulate a stable automotive network, we test with traffic loads generated from cyclic automotive applications such as updates sent between actuators and sensors, cyclic polling graphic updates, and fast diagnostic data for a drive that produces samples every few milliseconds. In total, 17 cyclic and periodic communication streams are generated within different endpoints. Data size is fixed and remains constant along simulating. The features of the streams are given in Table. 2:

The streams are assigned with two priorities, and the payload sizes vary from applications. High-priority streams (f_H) are typically under 1000 bytes, and the interval times are usually measured in milliseconds. The payload size of low-priority streams (f_L) varies from 2 to 1500 bytes and generates a faster rate measured in microseconds.

During the initialization of the simulations, the bandwidth of all the streams is registered at all gateways using SRP. The bandwidth is calculated as: *Bandwidth = second/interval* × *(framesize + SRP_SAFETYBYTE + PREAMBLE_BYTE + SFD_BYTES)* × *8 + INTERFRAME_GAP_BITS*. Streams with a shorter frame size than the minimum Ethernet frame will be padded to 64 bytes for transmission. The worst-case E2E latency of different streams are collected to measure the performance of the shaper.

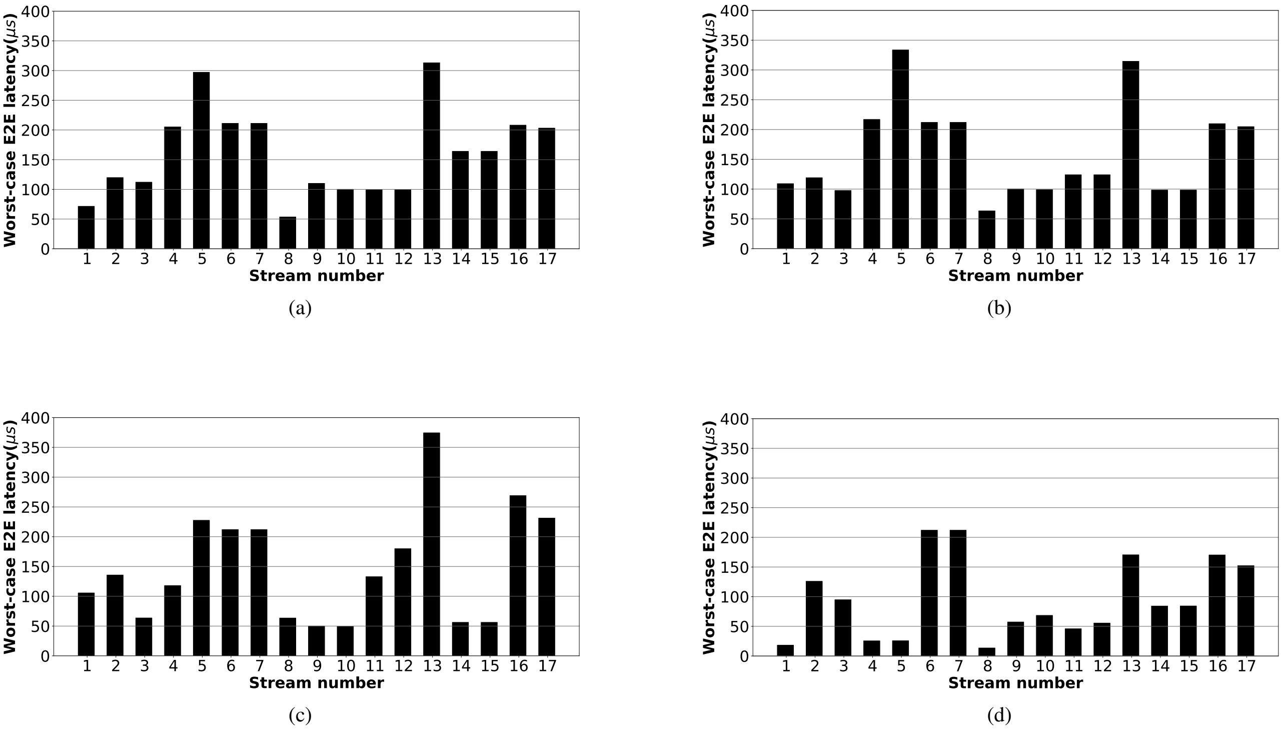


Fig. 5. A comparison of the worst-case E2E latency of all the testing streams in four scenarios: (a) Scenario 1, implementing standalone CBS for all the streams in the testing model, (b) Scenario 2, implementing CBS and strict priority for all the streams in the testing model, (c) Scenario 3, implementing TAS for all streams and an additional CBS for low-priority streams in the testing model, and (d) Scenario 4, implementing ATS and strict priority for all streams in the testing model.

C. Scenarios

Based on the same automotive Ethernet topology, we implement four sets of simulations using different shapers and schedulers for parallel comparison. Streams are matched with shapers by assigning traffic class/priority.

- In scenario 1, all the streams are assigned with the same traffic class and shaped by a CBS.
- In scenario 2, all the streams are shaped by a CBS. Then the transmitter select frames by strict priority. We use two traffic classes in the gateways and endpoints, namely AVB service reservation (SR) class A and class B for high and low priority streams, respectively.
- TAS is implemented in scenario 3. ST class is assigned to the high priority streams. GCLs are configured offline based on the pre-knowledge of the transmitting timing and cycle length of the streams. The low priority streams are assigned with AVB SR class A and are shaped by a CBS. The f_L streams will be transmitted only while the transmission gates for the f_H streams are closed.
- In scenario 4, we use ATS and strict-priority traffic selection for two traffic classes, referring to the high-priority and low-priority streams in Table 2.

D. Test results

The results of all the test cases are given in Fig. 5. Comparing Fig. 5(a) with Fig. 5(b), we observe that in the second scenario, when CBS combines with strict-priority, all the f_H streams ($f_H \in f_{3,9,10,14,15}$) have shorter E2E latency than they are in the first scenario. On the contrary, the f_L streams ($f_L \in f_{1,2,4,5,6,7,8,11,12,13,16,17}$) have longer latency when the other streams are prioritized.

Fig. 5(c) shows the results of using TAS and CBS. We found a noticeable influence on the streams, where the f_H streams have shorter latency than in scenario 2. In comparison, the f_L streams have longer or similar latency values. As we discuss in the previous sections, TAS opens the transmission window for the f_H streams when they arrive. Additionally, to ensure all the f_H frames are transmitted successfully, the length of the gate opening period of the ST queue is set with an extra margin. This can cause an extra queuing delay for the f_L streams that share the same ports with the f_H traffics. Due to usingthe CBS and strict-priority can not avoid all the interference from the f_L streams, in scenario 2, the f_H streams can be transmitted before the frames of the f_L streams that are currently queued, but still need to wait for any ongoing transmission of the f_L frames. From the results, we observe that TAS provides a more efficient and faster transmission to high-priorities than CBS and strict-priority. The latency differences of both traffic classes in Fig. 5(b) and Fig. 5(c) reflect that TAS delays the f_L streams to grant a shorter latency to the f_H streams.

Fig. 5(d) shows the scenario of using ATS and strict-priority. As we can see, the f_H streams have a shorter latency than some of the f_L streams. However, $f_{L1} \in f_{1,4,5,8,11,12}$ streams achieve shorter latency than the f_L streams. We found the common feature in f_{L1} is that all the streams have shorter frames than the other streams. And since ATS executes a per-class shaping manner, f_{L1} streams are transmitted with short delay when the *burstSize* is set with an adequate amount (10 consecutive frames for f_{L1} streams in our case). The results also show a significantly decreased latency to most f_L streams, except $f_{L2} \in f_{2,6,7}$, which remain at a similar scale among all four scenarios. The f_{L2} streams are transmitted within the Infotainment Domain and with large frame size. And utilizing ATS for f_{L2} streams has a limited influence on the worst-case latency since we set the *burstSize* for f_{L2} streams equals to double the frame size. We also found that all f_H streams have slightly increased latency compared with TAS with CBS in scenario 3, as a trade-off for the decreased latency of f_L.

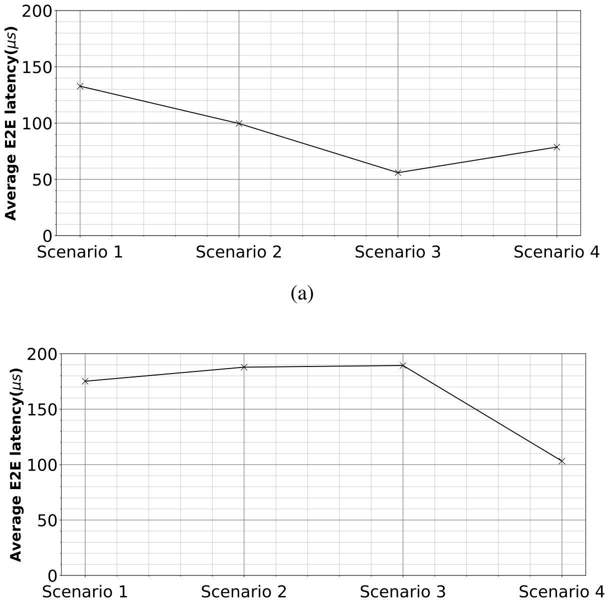
Fig. 6 shows the average latency comparison among all four scenarios. The results in Fig. IV.D indicate that TAS ensures the shortest latency to all the f_H streams. Meanwhile, ATS also guarantees a shorter latency to the f_H streams than CBS and strict-priority. In contrast, we can see in Fig. IV.D that ATS provides a significantly shorter average latency to the f_L streams than the other cases. At the same time, TAS has the longest latency for f_L, which is slightly longer than CBS and strict-priority. The results in Fig. 6 confirm that TAS offers shorter latency to high-priority class traffics than ATS in our simulated environment. One of the reasons is the f_H streams in the simulations are generated periodically, and the gate control actions are created accordingly. Therefore in scenario 3, all frames belonging to the f_H streams experience a very short delay at every node.

Because of the per-class shaping manner of ATS, the f_L streams in scenario 5 perform better than in TAS. On the other hand, the f_H streams do not degrade remarkably than in TAS. Which is due to the utilization of strict-priority at the egress ports, frames of the f_H streams are picked prior to those of f_L streams when they come out from the shaped queues.

V. CONCLUSION AND FUTURE WORK

Automotive Ethernet may replace other IVN communication technologies and keep a single physical network for applications [9]. One distinct benefit is reducing the amount of wiring harness since multiple domains share the same wire instead of different dedicated wires. Furthermore, the versatile TSN tools ensure the stringent latency requirement over Ethernet.

In this paper, we analyze using TSN scheduling and shaping as a component for future automotive Ethernet. After giving the features of the automotive network based on an example topology, we explain two TSN scheduling and shaping approaches: TAS and ATS in detail. We have discussed that using TAS and ATS in automotive Ethernet can achieve certain performance for automotive applications while also having specific restrictions. TAS scheduling is able to guarantee bounded short delay, but it requires acknowledgment of streams to set up the synchronous transmission. ATS shaping provides more flexibility of traffics but less determinism than TAS. Finally, we propose testing models for ATS and TAS gateway and have implemented a series of simulations within the OMNeT++ simulation framework in order to study the worst-case latency performance of TAS and ATS in IVN.



(a)

(b)

Fig. 6. A comparison of average E2E latency of the same traffic class in four scenarios: (a) Average E2E latency of all f_H streams and (b) Average E2E latency of all f_L streams.

The TSN-based automotive Ethernet aims to provide a deterministic guarantee for mixed-type traffic loads. Based on our results, TAS can offer shorter worst-case E2E latency for periodic high-priority streams than the other scheduling schemes in our IVN scenario. Applications such as control flow from chassis, which are usually periodic and have stringent latency requirements, can benefit from being set as ST traffic class. At the same time, low-priority streams may have longer E2E latency using TAS in the gateways than the other shapers, however suitable for the profile of applications which have tolerant requirement on latency, for instance, system management and configuration traffic.

ATS has the second shortest latency performance for high-priority streams in the tests but achieves better average latency for all streams. It is a preferred choice when the requirements of time-sensitive streams can be fulfilled and also accomplish certain transmission latency for low-priority streams, e.g., video and voice traffics. Compared with the mechanism of TAS, ATS is enabled to assign the higher priority class to both sporadic and periodic streams.

From the perspective of synthesis requirements, both ATS and TAS need the flow classification configuration at the design time. It is feasible to assign traffic classes manually in some cases. Otherwise, a synthesis tool can be used [34]. On the other hand, gate operations in TAS at every node also need to be set up, which requires more effort to build the network configuration than ATS.

Additionally, our proposed testing models provide a wide range of configurable attributes. In this paper, the testing results are generated based on a simulated IVN configuration. We have verified the functionalities of the models from all simulations.

Moreover, it is possible to scale up the model with multiple setups and compatible with a specific network environment. We expect to use the models on a real-world use case in the future.

We currently test with periodic flows in the simulation, future work may focus on the impact of using TAS and ATS scheduling for sporadic flows. Furthermore, future simulation should also include varying traffics under different shaper configurations. Additionally, the formula of ATS delay bound is rather conservative. It seems infeasible to predict the performance in IVN theoretically. A combined theoretical and practical analysis may offer a closer approximation of the latency bound. Addressing these problems is important to offer TSN functionalities to automotive Ethernet.

REFERENCES

[1] L. L. Bello, R. Mariani, S. Mubeen, and S. Saponara. “Recent advances and trends in on-board embedded and networked automotive systems,” *IEEE Trans. Industrial Informatics* vol. 15, no. 2, pp. 1038–1051, Feb. 2018.

[2] L. L. Bello, and W. Steiner. “A perspective on IEEE Time-sensitive networking for industrial communication and automation systems,” *Proc. IEEE* vol. 107, no. 6, pp. 1094–1120, June 2019.

[3] R. Bosch, “CAN specification version 2.0,” pp. 2–68, 1991.

[4] F. Hartwich, “CAN with flexible data-rate,” in *Proc. IEEE ICC*, 2012.

[5] M. Ruff, “Evolution of local interconnect network (LIN) solutions,” in *Proc. IEEE VTC Fall*, 2003.

[6] Cooperation, M. O. S. T. “MOST-media oriented systems transport,” MOST Specification Rev 3 (2005): E2.

[7] R. Makowitz and C. Temple. “Flexray-a communication network for automotive control systems,” in *Proc. IEEE International Workshop on Factory Communication Systems*, 2006.

[8] D. Porter, “100BASE-T1 Ethernet: The evolution of automotive networking,” Texas Instruments, Techn. Ber 2018.

[9] L. L. Bello, “Novel trends in automotive networks: A perspective on Ethernet and the IEEE audio video bridging,” in *Proc. IEEE ETFA*, 2014, pp. 1–8.

[10] IEEE P802.1DG, “TSN Profile for Automotive In-Vehicle Ethernet Communications”. Oct. 2019.

[11] IEEE 802.1Qbv–2015, “IEEE standard for local and metropolitan area networks—bridges and bridged networks—amendment 25: Enhancements for scheduled traffic”, pp. 1–57, Mar. 2016.

[12] IEEE P802. 1Qcr, “IEEE Draft standard for local and metropolitan area networks–media access control (MAC) bridges and virtual bridged local area networks amendment: Asynchronous traffic shaping”. Aug. 2019.

[13] A. Nasrallah *et al.*, “Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS),” IEEE Access vol. 7, pp. 44165–44181, Apr. 2019.

[14] A. Grigorjew, F. Metzger, T. Hoßfeld, and J. Specht. “A simulation of asynchronous traffic shapers in switched ethernet networks,” in *IEEE Proc. NetSys*, 2019.

[15] IEEE 802.1Qci, “Draft standard for local and metropolitan area networks bridges and bridged networks amendment: Per-stream filtering and policing,” Sept. 2017.

[16] M. Böhm, J. Ohms, O. Gebauer, and D. Wermser. “Architectural design of a TSN to SDN gateway in the context of in-dustry 4.0,” 2018.

[17] T. Hackel, P. Meyer, F. Korf, and T. C. Schmidt. “Software-defined networks supporting time-sensitive in-vehicular communication,” in *Proc. IEEE VTC*, 2019.

[18] J. Specht and S. Samii. “Urgency-based scheduler for time-sensitive switched ethernet networks,” in *Proc. IEEE ECRTS*, 2016.

[19] A. Varga, “The OMNeT++ discrete event simulation system”. in *Proc. ESM*, 2001.

[20] T. Steinbach, H. D. Kenfack, F. Korf, and T. C. Schmidt, “An extension of the OMNeT++ INET framework for simulating real-time Ethernet with high accuracy,” in *Proc. ICST*, 2011.

[21] IEEE 802.1Qat, “IEEE standard for local and metropolitan area networks—bridges and bridged networks—amendment 25: Stream reservation protocol (SRP)”, June 2010.

[22] Z. Zhou, M. S. Berger, S. R. Ruepp, and Y. Yan. “Insight into the IEEE 802.1 Qcr asynchronous traffic shaping in time sensitive network,” *Advances Science, Technol. Engineering Systems J.* vol. 4, no. 1, pp. 292–301, 2019.

[23] T. Steinbach, “Ethernet-based network architectures for future real-time systems in the car”, *ATZ Worldw*, vol. 121, no. 7, pp.72–77, 2019.

[24] J. Migge, J. Villanueva, N. Navet, and M. Boyer. “Insights on the performance and configuration of AVB and TSN in automotive Ethernet networks,” in *Proc. ERTS*, 2018.

[25] D. Thiele, R. Ernst, and J. Diemer, “Formal worst-case timing analysis of Ethernet TSN’s time-aware and peristaltic shapers,” in *Proc. IEEE VNC*, 2015.

[26] L. Zhao, P. Pop, and S. S. Craciunas, “Worst-case latency analysis for IEEE 802.1 Qbv time sensitive networks using network calculus,” IEEE Access, vol. 6, pp. 41803–41815, July 2018.

[27] S. Thangamuthu, N. Concer, P. J. Cuijpers, and J. J. Lukkien. “Analysis of ethernet-switch traffic shapers for in-vehicle networking applications,” in *Proc. IEEE DATE*, 2015.

[28] P. Meyer, T. Steinbach, F. Korf, and T. C. Schmidt. “Extending IEEE 802.1 AVB with time-triggered scheduling: A simulation study of the coexistence of synchronous and asynchronous traffic,” in *Proc. IEEE VNC*, 2013.

[29] J. H. Kim, S. Seo, N. Hai, B. M. Cheon, Y. S. Lee, and J. W. Jeon. “Gateway framework for in-vehicle networks based on CAN, FlexRay, and Ethernet,” *IEEE Trans. Veh. Technol.* vol. 64, no. 10, pp. 4472–4486, Nov. 2014.

[30] Y. S. Lee, J. H. Kim, and J. W. Jeon. “FlexRay and Ethernet AVB synchronization for high QoS automotive gateway,” *IEEE Trans. Veh. Technol,* vol. 66, no. 7 pp. 5737–5751, Dec. 2016.

[31] P. Meyer, F. Korf, T. Steinbach, and T. C. Schmidt. “Simulation of mixed critical in-vehicular networks,” *Recent Advances Network Simulation*, pp. 317–345. Springer, 2019.

[32] K. Müller, T. Steinbach, F. Korf, and T. C. Schmidt. “A real-time Ethernet prototype platform for automotive applications,” in *IEEE ICCE*, 2011.

[33] L. Zhao, F. He, E. Li, and J. Lu, “Comparison of time sensitive networking (TSN) and TTEthernet,” in *Proc. IEEE/AIAA DASC*, 2018.

[34] J. Specht and S. Samii. “Synthesis of queue and priority assignment for asynchronous traffic shaping in switched ethernet,” in *Proc. IEEE RTSS*.

Zifan Zhou received his M.Sc. in Telecommunication Engineering from the Technical University of Denmark (DTU) in 2018. And currently he is a Ph.D. candidate at the Network Platform and Service Research Group studying Latency-Critical Transmission, his research focuses on simulations and hardware design of ultra-low latency networks.

Juho Lee received the B.S. degree in Computer Engineering from Sejong University, Seoul, Korea, in 2011. He is currently working toward the unified course of M.S./Ph.D. degree with the Department of Electronics and Computer Engineering, University of Hanyang, Seoul, Korea. His current research deals with in-vehicle network (such as Ethernet AVB and TSN), V2X communication and future networks.

Sungkwon Park (Senior Member, IEEE) received the B.S. degree from Hanyang University, Seoul, South Korea, in 1982, the M.S. degree from the Stevens Institute of Technology, Hoboken, NJ, USA, in 1983, and the Ph.D. degree from the Rensselaer Polytechnic Institute (RPI), Troy, NY, USA, in 1987, all in Electrical Engineering. He joined Tennessee Technological University, Cookeville, TN, USA, as an Assistant Professor with the Department of Electrical Engineering, upon his graduation from RPI, in 1987. He left Tennessee Tech, in 1993, as a tenured Associate Professor, to join Hanyang University. He is currently a Full Professor with the Electronic Engineering Department and the Dean of the College of Engineering, Hanyang University. He has published nine authored books and over 250 refereed technical articles in the areas of CATV, IPTV multimedia systems, and vehicle communication systems. He holds over 45 domestic and international patents. His research interests include in-vehicle networks, in-robot networks, and autonomous vehicles. He is a Life-Long Member of the Korean Information and Communications Society. He received the Initiation Award from the U.S. National Science Foundation, in 1990 and a prime minister award of Korea in 2011.

Michael Stübert Berger was born in 1972 and received the M.Sc. EE and Ph.D. from the Technical University of Denmark in 1998 and 2004. He is currently Associate Professor at the university within the area of switching and network node design. Michael Berger has been participating in several projects with relation to TSN. He was Project Leader on the national project ERAN (Ethernet for RAN) where TSN was explored in the Front haul Network. Furthermore, he coordinated the participation from DTU in a Eurostars Project on TSN (Fronthaul for CRAN). He is currently responsible for the Department participation in a Nordic University HUB projet on Industrial IoT, Fog computing and TSN and he is currently Mentor of 1 Post Doc and 2 PhD students in the area of TSN and deterministic networks.

Ying Yan received the B.Eng. degree in Electrical Engineering from the Beijing University of Technology, China, in 2002. She has received her M.S. degree in Electronics Engineering in 2004, and the Ph.D. degree in Telecommunication Engineering in 2010 from Technical University of Denmark. During 2006–2007, she worked as a Research Scientist at the department of com-munication platforms in the Technical Research Centre of Finland (VTT), Finland. Her research fields include time sensitive network (TSN), Internet of things (IoT) network, 5G mobile network, network simulation and emulation, traffic data mining and analysis, and network security.