DEU FMD 26(76) (2024) 105-118

Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi
Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering
Elektronik/Online ISSN: 2547-958X

RESEARCH ARTICLE / ARAŞTIRMA MAKALESI

Cyber Security for IEEE 802.1 Time Sensitive In-Vehicle Networking: Recent Advances and Impact Analysis of DoS Attacks

IEEE 802.1 Zaman Hassas Araç-İçi Haberleşme Ağları için Siber Güvenlik: Güncel Gelişmeler ve Hizmet Reddi Saldırılarının Etki Analizi

Mustafa Topsakal [1,2], Selçuk Cevher [1*]

[1] Department of Computer Engineering, Faculty of Engineering, Karadeniz Technical University, Trabzon, Turkey
[2] Department of Software Engineering, Faculty of Technology, Karadeniz Technical University, Trabzon, Turkey

Corresponding Author / Sorumlu Yazar *: cevhers@ktu.edu.tr

Abstract

Dedicated computers control the operation of Cyber Physical Systems such as in-vehicle architectures with the help of sensors and actuators. Securing the real-time communication technologies for in-vehicle platforms attracts the attention of the research community. IEEE 802.1 Time Sensitive Networking (TSN) task group targets the standardization of Ethernet-based deterministic communication technologies due to its high bandwidth and low cost. IEEE P802.1DG working group specifies the profile for secure and real-time automotive embedded systems relying on TSN, which is envisioned to be widely used in future in-vehicle platforms. In this paper, we present an elaborate review of the research work on the security of in-vehicle communication networks with TSN support, and define various Denial of Service (DoS) attack scenarios targeting the real-time traffic in in-vehicle networks. We evaluate the impact of our attack scenarios on the performance of two different realistic in-vehicle communication networks with varying sizes. Experimental results show that DoS attacks can lead to severe consequences disrupting the healthy operation of safety-critical functions in a TSN-compliant in-vehicle network.

Keywords: *In-Vehicle Network, Deterministic Real-Time Communication, IEEE 802.1 TSN, DoS Attacks*

Öz

Adanmış bilgisayarlar, araç-içi mimariler gibi Siber Fiziksel Sistemlerin operasyonunu sensörler ve aktüatörler yardımıyla kontrol ederler. Araç-içi platformlar için gerçek-zamanlı haberleşme teknolojilerinin güvenli hale getirilmesi araştırma topluluklarının dikkatini çekmektedir. IEEE 802.1 Zaman Hassas Haberleşme (ZHH) görev grubu, yüksek bant genişliği ve düşük maliyeti nedeniyle, Ethernet tabanlı deterministik iletişim teknolojilerinin standardizasyonunu hedeflemektedir. IEEE P802.1DG çalışma grubu, gelecek nesil araç-içi platformlarda yaygın olarak kullanılması öngörülen ZHH teknolojisine dayalı, güvenli ve gerçek-zamanlı otomotiv gömülü sistem profili tanımlamaktadır. Bu makalede, ZHH'yı destekleyen araç-içi haberleşme ağlarının güvenliği ile ilişkili araştırma çalışmalarının kapsamlı bir incelemesi sunulmakta ve araç-içi haberleşme ağlarındaki gerçek zamanlı trafiği hedef alan Hizmet Reddi (HR) saldırı senaryoları tanımlanmaktadır. Saldırı senaryolarımızın, değişken boyutlara sahip iki farklı, gerçekçi araç-içi haberleşme ağlarının performansı üzerindeki etkisi analiz edilmektedir. Deney sonuçlarımız, HR saldırılarının ZHH uyumlu bir araç-içi haberleşme ağındaki güvenlik-kritik fonksiyonların sağlıklı işlevini kesintiye uğratacak ciddi sonuçlara yol açabileceğini göstermektedir.

Anahtar Kelimeler: *Araç-İçi Ağlar, Deterministik Gerçek-Zamanlı Haberleşme, IEEE 802.1 TSN, Hizmet Reddi Saldırıları*

1. Introduction

Physical processes in Cyber Physical Systems (CPS) such as in-vehicle platforms are managed by control computers with the help of sensors and actuators [1]. The evolution of sensor hardware and autonomous control facilities such as *Advanced Driver Assistance Systems* (ADAS), and automated driving lead to an increase in the number of electronic components within a vehicle [2]. These interconnected components have stringent timing requirements, which necessitate the deployment of a deterministic real-time in-vehicle communication network with bounded transmission delay and jitter [3]. The real-time communication among the vehicle components should be protected against cyber-attacks since the increased connectivity as well as the number of components within a vehicle expose a higher amount of attack surfaces [4, 5]. For example, the ability to remotely update the software run by an *Electronic Control Unit* (ECU) in an in-vehicle communication platform is an example for potential security vulnerabilities [6].

IEEE 802.1 Time Sensitive Networking (TSN) task group aims at establishing the standard specifications for deterministic real-time variants of Ethernet technology. In order to leverage the important features of TSN for in-vehicle networking, P802.1DG [7] specifies the profile for secure and real-time automotive embedded systems relying on TSN. IEEE 802.1 TSN standard provides a mixed-time-criticality capability by supporting the traffic types of *Time-Triggered* (TT) with strict timing requirements, *Audio-Video Bridging* (AVB) with soft real-time constraints and *Best-Effort* (BE) without any timing concerns [8].

DOI:10.21205/deufmd.2024267613

Atıf şekli / How to cite:
Topsakal, M., Cevher, S. (2024). Cyber Security for IEEE 802.1 Time Sensitive In-Vehicle Networking: Recent Advances and Impact Analysis of DoS Attacks. DEUFMD, 26(76), 105-118.

Geliş Tarihi / Received: 20.03.2023
Kabul Tarihi / Accepted: 25.07.2023

IEEE 802.1Qbv aims at prioritizing the forwarding of hard real-time TT traffic by utilizing a scheduling table called *Gate Control List* (GCL) for each outgoing port, which determines the time offsets to place the network frames onto a transmission line. Each AVB stream transmitted in the network belongs to either *Class A* or *Class B* such that Class A streams have a higher priority than Class B traffic. The transmission of an AVB frame over an outgoing link is enabled only if no higher-priority frame awaits to be transmitted over the same link. Using the prior knowledge of stream properties, the real-time communication technologies for CPS environments can be configured at design time in advance of the actual network operation [9].

Since TSN is foreseen to be widely used in safety-critical in-vehicle platforms, it is increasingly crucial to enhance TSN to address the security vulnerabilities [7]. An attacker manipulating the properties of real-time streams may significantly influence the network latency of the victim streams, and, hence, disrupt the deterministic real-time communication [5, 10]. One way of intruding an in-vehicle communication is to realize *Denial of Service* (DoS) attacks, where an attacker may inject forged messages into the network, resulting in reduced bandwidth, network congestion, and prolonged delays for real-time traffic streams [11]. In this paper, we present a thorough review of the literature on the security in TSN and specifically the research work on the security of in-vehicle communication networks. We define various DoS attack scenarios targeting in-vehicle networks by manipulating the message frequency and payload size of the compromised AVB traffic. We also realize attack scenarios compromising the TT traffic by manipulating the gate mechanism specified by 802.1Qbv. We evaluate the impact of our attack scenarios on the performance of two different realistic in-vehicle communication networks with varying sizes from [12, 13, 14] and [15] relying on TSN for real-time data delivery. The impact of each attack scenario on the delay characteristic of real-time streams is evaluated in terms of *Worst-Case Delay* (*WCD*) and the number of unschedulable AVB streams not satisfying the deadline constraints. The WCD of an AVB stream can be computed via mathematical tools by finding the sum of latency bounds in network nodes along the route of the stream [16]. Our attack scenarios are implemented by extending the infrastructure provided by [17] to build our traffic scenarios, manipulate the properties of the compromised streams, and perform the *WCD* analysis for the AVB transmissions according to the *AVB Latency Math* tool specified in the IEEE 802.1BA standard. Our paper is the extended version of the research work presented in [18] which relies on a limited number of traffic scenarios with homogeneous-criticality support. Taking into consideration that a considerably limited amount of research exists in the field of security in in-vehicle embedded systems with TSN support [2], we believe that our paper significantly contributes to the literature as follows:

- It provides an elaborate review of the current state of the art especially in the field of security of in-vehicle networks with TSN support.
- In addition to the DoS attack scenarios targeting TSN-based in-vehicle communication platforms with homogeneous-criticality support, where solely the AVB streams are transmitted, it defines various realistic DoS attack use cases manipulating the TT streams in an in-vehicle platform with mixed-criticality support, and studies the impact of these use cases on the delay of lower-priority AVB streams.
- It investigates the impact of the attack scenarios on the performance of realistic in-vehicle platforms by exploiting two different in-vehicle communication networks with varying sizes from the literature, one of which is provided by a major motor company in China [15].

The rest of this paper is organized as follows. Section 2 presents our motivation and related work. Section 3 describes the IEEE 802.1 TSN communication technology. Sections 4 and 5 present the system model and an example operation of DoS Attacks for in-vehicle networks with TSN support, respectively. Section 6 reports our experimental results while Section 7 provides a discussion and plans for future work. Finally, Section 8 concludes the paper.

2. Motivation and Related Work

The increasing number of electronic components within in-vehicle communication networks in automotive embedded systems along with their increased connectivity with pedestrians and other vehicles widens the surface exposed to cyber-attacks. The conventional in-vehicle networking technologies such as Controller Area Network (CAN) and FlexRay are not designed with security in consideration, and, hence, are characterized by several security vulnerabilities in terms of confidentiality, authentication and availability [6]. CAN is the most widely used technology for in-vehicle communication, designed by Bosch to multiplex communication between ECUs in a vehicle through a bus, and thus to decrease the overall wiring cost [19]. The broadcast nature of CAN and FlexRay relying on a shared bus poses a risk for confidentiality since a compromised ECU can be used to eavesdrop the data communication, while the lack of a signature mechanism within the data link layer of CAN introduces an authentication vulnerability. Furthermore, the multi-master feature of CAN risks the availability of the embedded system since a malicious master ECU can perform an injection attack by continuously sending forged messages making the bus unavailable to the other ECUs.

2.1. Security in TSN

In order to leverage the important features of TSN for in-vehicle networking, P802.1DG [7] specifies the profile for secure and real-time automotive embedded systems relying on IEEE 802.1 TSN, which should be further enhanced to address the aforementioned security limits of conventional in-vehicle networks. Thanks to the increased data rates in TSN, the critical data to be transmitted can be efficiently encrypted. However, this encryption may cause the end-to-end communication delays of data streams to grow due to the extra processing required by the encrypted data, possibly violating the timing requirements of network applications. Therefore, the overall network traffic can be categorized into security classes with varying priorities such that only the data with the highest priority is encrypted [6]. Furthermore, TSN can be combined with an intrusion detection mechanism, where the features of the incoming traffic can be extracted, and, then, classified to assess the normal behavior of the network by using an initial training phase. For this purpose, the issue of clock skew regarding the time synchronization should be additionally considered. A proposal from Bosch for an intrusion detection system targeting TSN has been recently presented in [20]. Ergenç et. al. [21] and Bello and Steiner [22] highlight that ensuring security is a crucial design issue for TSN by emphasizing that more research is needed on security in TSN including the early detection of cyber threats. Maximizing security in TSN under time-constrained conditions formulated as an optimization problem, Mahfouzi et al. [23] takes the time duration required by the tasks of encryption/decryption into consideration to compute the routes and scheduling tables for the data streams. In order to satisfy the real-time, safety and security requirements, Reusch et al. [24] jointly addresses the safety and

security requirements in TSN. For this purpose, they propose three different methodologies including a constraint-programming formulation, Simulated Annealing meta-heuristic and a heuristic approach to solve the problems of task scheduling, and the computation of redundant disjoint routes and scheduling tables for data streams. While Wüsteney et al. [25] analyzes the latency and jitter caused by the usage of a firewall and packet filtering in a TSN network, Pena et al. [26] studies the impact of using MACSec in TSN on the timing requirements of specific applications. Emphasizing that MACSec is a suitable solution to secure IEEE 802.1AS time synchronization protocol of TSN, Tang et al. [27] models 802.1AS using hierarchical color Petri Nets, and carries out a simulation based on the protocol model with MACsec.

Since the central architecture of Software Defined Networking (SDN) paradigm may be beneficial to increase the security level in a network, TSN and SDN technologies can be combined in order to enhance the security aspect of real-time communication [28]. Li et al. [29] proposes a methodology, which relies on SDN/NFV (Network Functions Virtualization) combining network state and time synchronization information to detect and mitigate the attacks against *Precision Time Protocol*. Underlining that TSN and SDN can be used in combination, Böhm et al. [30] presents a gateway architecture providing admission control for the traffic exchanged between TSN and SDN networks, observing the security requirements of real-time applications.

2.2. Cyber Attacks Against In-Vehicle Networks

Focusing on the in-vehicle attack surfaces such as on-board diagnostic ports (OBD-II), media systems, and short/long range wireless interfaces, the work in [31] underlines that an attacker can exploit the internal buses of a vehicle via physical or wireless access methods. It further investigates the possible violation use cases specifically for media player, OBD-II, and telematics devices within a vehicle. Using real in-vehicle experimental setups, the studies of [32, 33, 34] demonstrate that cyber-attacks against CAN bus within a vehicle can be performed via OBD-II port by injecting malicious messages into network, and these attacks may significantly threaten the correct operation of a vehicle. The survey presented in [35] emphasizes that an automobile may contain several ECUs which need to communicate among each other and with the outside world, and may be subject to different attack scenarios. The authors evaluate several real vehicle platforms to show how the attack surfaces have evolved, and recommend different defensive strategies to detect and prevent cyber-attacks against vehicles. The work in [36] explores the potential attack surfaces for Jeep Cherokee by investigating its network architecture, and carries out a remote attack against this vehicle. The authors show that the remote attack results in certain physical systems such as steering and braking to be affected by the realized attack scenario. Yan et al. [37] investigates the possible attacks against three fundamental types of sensors, namely ultrasonic sensors, millimeter-wave radars, and cameras, highlighting that these sensors are also used by the Autopilot systems in Tesla vehicles. The authors emphasize that jamming and spoofing attacks targeting these components can disrupt the operation of a vehicle. Zeng et al. [38] analyzes the impact of spoofing attacks on driving experience by providing road navigation systems within a vehicle with spoofed GPS inputs in order to mislead the driver towards a wrong route. The recent reports published by Keen Lab reveal that several defects exist in the designs of Tesla P75/P85/90D, BMW i3 94/X1 sDrive 18Li/525Li/730Li models, and Mercedes-Benz User Experience Platform, and the lack of strong cipher protection contributes to the possibility of cyber-attacks [39, 40, 41, 42]. These reports expose that unauthorized access to embedded components such as gateway, Head Unit, and Telematics is possible for aforementioned vehicles, and enabled the manufacturers to correct the vulnerabilities in their vehicles after the publication of the report. Table 1 presents the recent literature relevant to in-vehicle attack surfaces, which perform experiments in real in-vehicle platforms. It provides a comparison for the existing research work in terms of attacked surface of a vehicle, access type to compromise a vehicle, the way the attack is realized, and car model where the attack scenarios are implemented.

Using the small in-vehicle network topology shown in Figure 3a in this paper, Topsakal and Cevher [18] performs an impact analysis of DoS attack scenarios which manipulate solely the message period of the experimented AVB streams with homogeneous-criticality. Their experimental results show that a DoS attack can lead to severe consequences for the healthy operation of safety-critical functions in an in-vehicle network with TSN support. Our paper significantly extends the research work in [18] in the following aspects: i) Our work provides an elaborate review of the current state of the art in the field of security of time-sensitive in-vehicle communication networks whereas the literature review given in [18] is too limited, ii) As opposed to [18] relying on solely the traffic scenarios with homogeneous-criticality support, our paper exploits traffic scenarios with mixed-criticality support to define various realistic DoS attack use cases manipulating the TT streams in an in-vehicle platform, iii) Relying on the existing literature, our paper considers more realistic traffic scenarios in terms of a higher number of streams as well as more realistic stream properties corresponding to more heavily-loaded in-vehicle data communications, iv). Our study defines DoS attack scenarios with a higher diversity by additionally manipulating the payload size, GCL properties and the number of frames transmitted per second for the experimented traffic streams, and v). Our work investigates the impact of DoS attacks on the performance of a wider range of realistic in-vehicle communication networks with varying sizes including one provided by a major motor company in China [15].

2.3. Defense Strategies for In-Vehicle Networks

Bello et al. [43] specifies the defense techniques against cyber-attacks for in-vehicle networking, which are envisioned to exist within the next-generation vehicles. These techniques include the clustering of subsystems through the usage of gateways, embedding hardware accelerators in automotive computing units for encryption, using signature mechanisms for supervisory authentication, classifying ECUs into different security classes, and intrusion detection systems relying on physical or upper layer features. Underlining that Ethernet-based automotive networks such as TTEthernet and TSN provide a higher bandwidth and better timing guarantees, Lin and Yu [44] emphasizes that there is a trade-off between ensuring the safety and security of a vehicle. For this purpose, they investigate the security trends for the tasks of secret key management, frame replication and elimination, and Virtual Local Area Network (VLAN) segmentation for in-vehicle communication networks. Emphasizing that the original specification for CAN lacks a central monitoring mechanism, Häckel et al. [45] offers an architecture integrating TSN and SDN, in which all real-time traffic transmitted in an in-vehicle network is centrally monitored without increasing the network delay in order to prevent malicious traffic. Meyer et al. [46] performs anomaly detection by using TSN and SDN technologies together in an in-vehicle communication network, which relies on the filtering capabilities of *Per-Stream Filtering and Policing* (PSFP) specified

Table 1. Recent literature on in-vehicle cyber-attacks with experimentation on a real platform

Reference	Attack Surface	Access Type	Attack Realization	Experimental Platform
[32] Koscher et al. (2010)	OBD-II	Physical	Sending malicious messages into CAN bus	Real Car (model unspecified)
[31] Checkoway et al. (2011)	Media Player, Bluetooth, Telematics, Gateway	Physical, Short-long range wireless	Compromising some of the car's ECUs, and sending arbitrary messages to the communication network	Real Car (model unspecified)
[33] Miller and Valasek (2013)	OBD-II	Physical	Sending arbitrary messages into CAN bus	Ford Escape, Toyota Prius
[35] Miller and Valasek (2014)	TPMS, RKE, Telematics, Bluetooth, Radio, Wifi/Cellular	Wireless	Remotely accessing to some components and injecting malicious messages into CAN bus	Jeep Cherokee, Cadillac Escalade, Infiniti Q50 and others
[34] Woo et. al. (2015)	OBD-II	Long range wireless	Sending malicious messages into CAN bus	Real Car (model unspecified)
[36] Miller and Valasek (2015)	Telematics	Long range wireless	Injecting malicious messages into CAN bus	Jeep Cherokee
[37] Yan et al. (2016)	Sensors, Radars, Cameras	Wireless	Transmitting illegal data to sensors	Tesla Model S
[39] Nie et al. (2017)	Gateway, Parrot, IC	Short range wireless	Injecting malicious messages into CAN bus	Tesla Model S P75/P85
[38] Zeng et al. (2018)	GPS	Long range wireless	Manipulating road navigation systems with forged GPS inputs	Real Car (model unspecified)
[40] Nie et al. (2018)	Gateway, Body Control Modules, Autopilot ECUs	Short range wireless	Compromising the Autopilot module on the Tesla car and injecting malicious messages	Tesla Model S P85, Tesla Model X 90D
[41] Cai et al. (2019)	HeadUnit, Telematics, Central Gateway	Physical, Wireless	Injecting malicious messages into CAN bus	BMW i3 94/ 525Li/730Li/ X1 sDrive 18Li
[42] Keen Security Lab. (2021)	HeadUnit, Telematics	Wireless	Sending arbitrary messages into CAN bus	Mercedes-Benz User Experience Platform

by 802.1Qci through the configuration parameters such as stream/burst sizes and Maximum Transmission Unit (MTU) to differentiate between normal and abnormal network behaviors. Luo et al. [47] proposes an anomaly detection system for in-vehicle networks with TSN support relying on PSFP specified in 802.1Qci in order to block the messages exceeding a customized maximum service data unit size. The authors experimentally show that the proposed system successfully identifies four different abnormal traffic events in relation with DoS attacks. Meyer [48] presents an algorithm providing detection and prevention for DoS attacks against TSN switches targeting the CBS mechanism. Defining an attack scenario which fills the egress queues with arbitrary messages, and, hence, causes the CBS mechanism to block the legitimate frames, they also suggest the usage of 802.1Qci for the purpose of attack prevention.

Aoudi et al. [49] proposes an anomaly detection mechanism relying on the spectral analysis of CAN message payloads transmitted within a vehicle by addressing the real-world deployability challenges. Bozdal et al. [50] presents a wavelet-based approach to detect behavioural changes in the CAN network within a vehicle. Nowdehi et al. [51] proposes a detection mechanism for in-vehicle networks with CAN support relying on the analysis of message payloads. Please note that attack detection based on message payloads enables even the identification of complex intrusions which intentionally unchange the message frequency in order to harden the detection process for the network operator. Han et al. [52] proposes an approach to identifying abnormalities in the CAN network in a vehicle relying on the event-triggered message intervals. Defining four different attack scenarios, various machine learning models are utilized to extract the respective normal and abnormal behaviors of the network. Cho and Shin [53] propose an intrusion detection mechanism, where the message sending frequencies of ECU components are measured, and, based on this measurement, the normal network behavior is formulated through Recursive Least Squares (RLS) algorithm. In the study of Waszecki [54], lower and upper limits for message arrival curves are computed taking into consideration the message periods and jitter, and, based on these limits, the violations of expected arrival curves are detected.

Table 2 presents the recent literature relevant to in-vehicle defense strategies, which mostly focus on TSN and CAN as the underlying real-time communication technology. It provides a comparison for the existing research work in terms of communication technology in use, attack type, defense objective and strategy. In Table 2, the cases, where no information is provided by the respective publication regarding the corresponding column header, are specified by the term N/A (not available). Please note that the attack types listed in Table 2 are denoted by their well-known names in the literature.

3. IEEE TSN Standard

TSN provides a group of standards providing reliability, predictability and time synchronization for safety-critical automotive communications [6]. TSN leverages on the previous standards provided by the IEEE 802.1 working group on AVB including the Generalized Precision Time Protocol (802.1AS-2011) for time synchronization, Stream Reservation Protocol (802.1Qat) for bandwidth reservation within switches, and Credit-Based Shaper (CBS) (802.1Qav) for shaping AVB streams to prevent traffic bursts by relying on a credit value, and, hence, avoid the starvation of lower-priority BE messages. TSN extends the aforementioned suit of protocols by introducing novel standards, which are relevant to in-vehicle communications. For bounded low delay and jitter, TSN offers 802.1Q-2018 enrolling 802.1Qav/Qat, 802.1Qbv providing enhancements for scheduled

Table 2. Related works on defense strategies for in-vehicle networks

Reference	Real-Time Communication Technology	Attack Type	Defence Objective	Investigated Defence Strategy
Lin and Yu [44] (2016)	TTEthernet, TSN	N/A	Intrusion mitigation	Investigating the performance of the mitigations mechanisms such as secret key management, frame replication/elimination and VLAN segmentation
Meyer [48] (2016)	TSN	DoS	Intrusion detection/prevention	Detecting the intrusions based on the CBS mechanism, and preventing them relying on 802.1Qci
Cho and Shin [53] (2016)	CAN	Fabrication, Suspension, Masquerade	Intrusion detection	Differentiating between normal and abnormal network behaviour by formulating the normal through *Recursive Least Squares* algorithm
Waszecki et al. [54] (2017)	N/A	DoS, Timing Attacks	Intrusion detection	Determining the expected lower and upper limits for message arrival curves, and detecting the violations of these limits
Nowdehi et al. [51] (2019)	CAN	Suspension, Fabrication, Masquerade, Conquest	Intrusion detection	Detecting the intrusions by analyzing the message payloads
Meyer et al. [46] (2020)	TSN	DoS, DDoS	Intrusion detection/mitigation	Detecting the intrusions relying on the central architecture of SDN, and mitigating them based on the filtering capabilities of 802.1Qci
Luo et al. [47] (2021)	TSN	DoS	Intrusion detection/mitigation	Detecting and mitigating the intrusions relying on the capabilities on 802.1Qci
Aoudi et al. [49] (2021)	CAN	Suspension, Fabrication, Masquarade, Conquest	Intrusion detection	Detecting the intrusions through the spectral analysis of message payloads
Han et al. [52] (2021)	CAN	Flooding, Fuzzy, Replay	Intrusion detection	Detecting the intrusions relying on various machine learning models processing event-triggered message intervals
Bozdal et al. [50] (2021)	CAN	Replay, Suspension, DoS, Fuzzy and Spoofing	Intrusion detection	Detecting the intrusions through a wavelet-based approach
Hackel et al. [45] (2023)	TSN	N/A	Intrusion detection/prevention	Securing the network by proposing an in-vehicle SDN-based central network architecture

traffic, 802.1Qbu enabling the preemption of the lower-priority traffic by the higher-priority streams, 802.1Qch for cyclic queuing and forwarding, and 802.1Qcr-2020 for asynchronous traffic shaping. 802.1Qbv prioritizes the forwarding of TT traffic via a Time Aware Shaper, which opens or closes the gates at the front of egress queues according to port-specific GCLs containing a timetable for the data frames to be transmitted. To globally synchronize GCLs in the overall network, TSN provides 802.1ASrev improving the reliability of clock synchronization via replication mechanisms. For reliability, TSN includes 802.1CB duplicating frames and sending them over multiple disjoint routes for an increased probability of successful frame delivery as well as 802.1Qci for per-stream filtering and policing. For resource management, TSN offers 802.1Qcc extending the capabilities of 802.1Qat by allowing for more complex configurations with 802.1Qbv and preemption mechanisms via different network configuration methodologies. Finally, P802.1DG working group currently makes an effort to specify profiles for secure, highly reliable, deterministic latency, automotive in-vehicle switched Ethernet networks based on TSN.

Figure 1 demonstrates the architecture of 802.1Qbv-compliant TSN switch, featuring a maximum of eight queues with varying priorities for each outgoing port [55]. *Switching Logic* forwards a newly arrived frame to the corresponding egress queue based on its priority encoded within its frame header. A portion of the highest-priority egress queues are allocated to TT traffic while the remaining queues are assigned to AVB and BE frames in the order of decreasing priority taking into consideration that AVB frames have a higher precedence than BE traffic. *Transmission Selection Logic* enables the transmission of the highest-priority message awaiting in an available queue with an open gate, as long as CBS if present does not prevent it. The interference from the lower-priority traffic should be eliminated in order to guarantee the deterministic communication for TT traffic. This can be achieved by ensuring that a TT queue with an open gate is the only available queue while simultaneously keeping the gates of the remaining queues closed causing the lower-priority traffic to experience a higher delay.

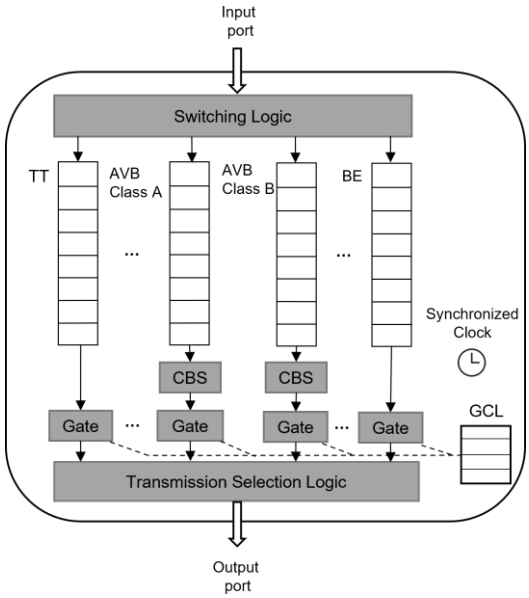
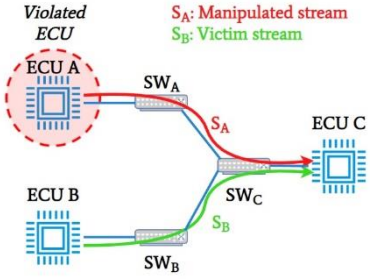
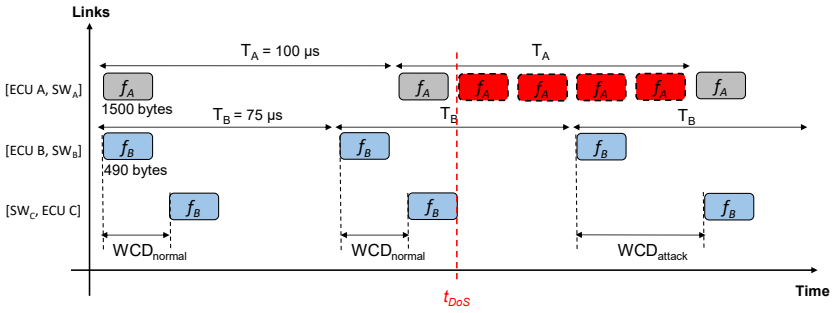


Figure 1. Structure of 802.1Qbv-compliant TSN Switch [56]



(a) Attack scenario



(b) Impact of a DoS attack

Figure 2. An example DoS attack scenario for an in-vehicle network with IEEE 802.1 TSN support

4. System Model

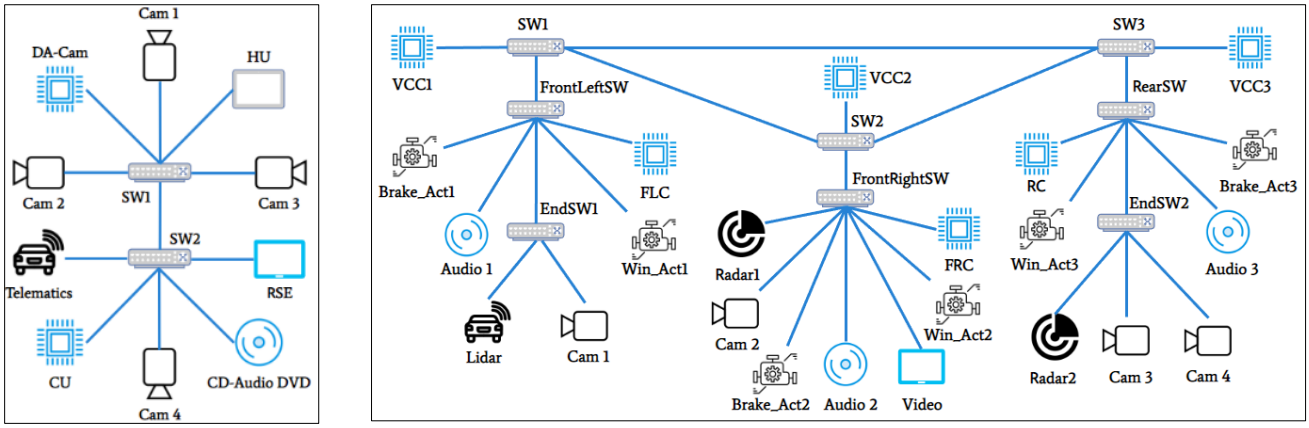
In this paper, the real-time AVB and TT traffic streams are assumed to be periodic. Each real-time stream is characterized by *deadline* (D), *message size* (S), *message period* (T) and *transmission route* (R) composed of source end-system (ES), intermediary switches and destination ES(s). Each AVB stream is classified into *Class A* or *B*, which has an influence on its timing requirements. Each egress port within network devices is associated with a scheduling table, namely GCL, which determines the timing of the open and close events for the gates located in front of the respective queues. In this work, only a single queue is assumed to be allocated within a network device for TT streams. Similar to [9] and [17], each TT stream is related with a set of GCLs, each of which is modelled by the parameters of *offset, period* and *duration*. These parameters specify the time offset at which the gate of the corresponding TT queue is opened, the period of the TT stream, and the time duration during which the gate is kept open, respectively. While the offset parameter for the first egress port traversed by a TT route is assumed to be user-supplied, the offset values for the subsequent GCLs along the route are obtained from the given offset. This derivation relies on the assumption that TT frames do not experience any queuing delay since each gate along the route is opened immediately after receiving a TT frame from the preceding node.

In this paper, we use an extended version of the AVB Latency Math tool from [17] to compute the worst-case delay for an AVB stream, which is capable of evaluating the impact of TT traffic on the lower-priority AVB traffic in addition to the interference from the other AVB streams. This tool determines a worst-case scenario for each link throughout the transmission route of the AVB stream subject to the analysis, which delays the stream the most. In order to identify such a scenario, it assumes that the AVB frame under analysis becomes available for transmission over a link when the bits of a TT frame have just started to be placed onto the same transmission line. Based on this assumption, it tries the start of each TT frame to find the position causing the maximum latency for the AVB stream. For each worst-case scenario, the extended tool evaluates the interference from the highest-priority TT streams in addition to the preemption overhead, which is then added to the interference from other AVB streams computed via the conventional AVB Latency Math tool. End-to-end WCD for an AVB stream is cumulatively determined by summing the WCDs for all links on its transmission route.

5. DoS Attacks in IEEE 802.1 TSN In-Vehicle Networks

DoS attacks aim at either disrupting the communication among ESs or causing the ESs to become unresponsive by consuming its resources [57]. The attacks targeting the real-time communication can be realized by manipulating the parameters of the compromised traffic streams including message period, number of frames per period, payload size or a combination of these features so that the ongoing transmissions sharing common links with the manipulated streams are impacted. DoS attacks realized by increasing the message frequency can be classified into *single* or *periodic*, which inject fabricated frames into the network only for once or periodically, respectively [58]. DoS attacks can result in serious safety issues in safety-critical in-vehicle platforms by causing the deterministic real-time communication to fail.

Figure 2 shows an example scenario for a TSN-compliant in-vehicle network with three ECUs and the impact of a DoS attack realized by increasing the message frequency on the ongoing real-time transmissions. As shown in Figure 2a, the violated *ECU A* and normally operating *ECU B* transmit lidar and radar data (S_A and S_B), respectively, destined to *ECU C*, which traverse some intermediary switches (SWs) and intersect at the link between SW_C and *ECU C*, namely [SW_C, *ECU C*], sharing the same priority queue. The respective message periods and payload sizes for S_A and S_B are shown in Figure 2b, where, at time t_{DoS}, an attacker compromises the stream S_A and initiates a DoS attack by injecting four forged frames coloured in red onto the link [*ECU A,* SW_A] within the message period T_A. For this experiment, the deadline



(a) Small topology **(b)** Large topology

Figure 3. Experimented in-vehicle communication networks

of S_A and S_B is selected to be 2 ms in compliance with the 802.1BA standard. As shown in the figure, up to t_{DoS}, the frames originating from *ECU B* arrive in *ECU C* with an end-to-end delay of $WCD_{normal} = 1.108$ ms, computed by the analysis tool described in Section 4, where the propagation delay is assumed to be zero. On the other hand, after t_{DoS}, the frames sourced at *ECU B* reach *ECU C* with an end-to-end delay of $WCD_{attack} = 2.517$ ms, which is greater than the deadline of 2 ms, showing the impact of the forged traffic on S_B. These results indicate that the violation of *ECU A* by the attacker increasing the message frequency of S_A causes the radar data carried by S_B to experience a much higher delay, and, hence, disrupting the operation of *ADAS* which compromises the driving safety of the vehicle. Please note that, for simplicity, only the frames belonging to S_B are displayed for the link [SW_C, *ECU C*] in Figure 2b.

6. Experimental Results

In this section, we define various DoS attack scenarios targeting in-vehicle platforms with TSN support, and evaluate their impact on the performance of two different realistic in-vehicle communication networks from [12, 13, 14] and [15]. In our experiments, one of the ESs in a network is assumed to be violated by an attacker using one of the methods described in Section 2.2. The attacker manipulates the properties of the AVB or TT streams originating from the compromised ES including message period, number of frames per period, and payload size. We also realize attack scenarios targeting the gate mechanism specified by 802.1Qbv, where the open duration of each gate located in front of the TT queue in each switch throughout the transmission route of the compromised TT stream is increased. This manipulation causes the lower-priority AVB traffic sharing common links with the TT traffic to experience a higher end-to-end transmission delay. The impact of each attack scenario on the delay characteristic of real-time streams is evaluated in terms of *WCD* and the number of unschedulable AVB streams not satisfying the deadline constraints. In our experiments, we also report the maximum link utilizations resulting from the attack scenarios, which is computed by determining the total amount of traffic passing through each network link divided by the available link capacity, and then selecting the link in the overall network with the maximum utilization. Each experimented in-vehicle network topology is modelled within an XML file. Similarly, each of our attack scenarios is implemented as an XML file, which contains the stream properties manipulated by the attacker, and is inputted to the infrastructure provided by [17] along with the XML file defining the network topology to perform the WCD analysis for each AVB transmission. Inspired by [17], the number of unschedulable AVB streams is computed as follows:

$$\sum_{S_i \in S_{AVB}} WCD(S_i) > S_i.D \qquad (1)$$

where S_{AVB}, $WCD(S_i)$ and $S_i.D$ represent the set of AVB streams transmitted in the network, the worst-case delay and deadline of the stream S_i, respectively. Please note that the summation term in Eq.1 corresponds to 1 if the condition holds, and 0 otherwise. As required by Eq. 1, *WCD* of an AVB stream S_i is evaluated by using the analysis tool from [17] explained in Section 4.

6. 1. Traffic Scenarios

In our experiments, we implement our attack scenarios using two different realistic in-vehicle communication networks with varying sizes from [12,13,14] and [15], which are demonstrated in Figure 3. Each of these experimented networks is composed of a different set of in-vehicle electronic components originating streams in order to actualize a realistic traffic scenario. Therefore, the number of streams transmitted in each network is necessarily not identical corresponding to use cases independent from each other. Figure 3a shows the small in-vehicle network topology with two SWs and 10 ESs, which is a well-known topology used in the literature [12,13,14]. This topology consists of two functional domains, namely *ADAS* and *Multimedia/Infotainment*. *ADAS* has a critical importance to ensure the vehicle safety, and is composed of the cameras *Cam1* to *Cam4*, *Head Unit* (HU) and two ECUs, namely *Control Unit* (CU) and *DA-Cam*. The compressed safety-critical visual data originating from *Cam1- Cam4* located in different parts of the vehicle is transferred to *DA-Cam*, which processes this data in order to generate an aggregated video stream for a bird-eye view and navigation alerts. The processed video and alerts from *DA-Cam* are transferred to *HU*, which gives visual assistance to the driver by visualizing the data from *DA-Cam* on a monitor. The alerts from *DA-Cam* are also sent to *CU*, which generates real-time control messages for both *DA-Cam* and *HU*. The *Multimedia/Infotainment* system of the in-vehicle scenario shown in Figure 3a contains a *CD-Audio/DVD* node which transmits audio and video streams to the *Rear Seat Entertainment* (RSE) to provide passengers with multimedia via monitors and speakers. The *Telematics* node transmits non-safety-critical real-time information such as GPS data to *HU* and *RSE*.

Table 3. Stream properties for the experimented in-vehicle communication networks

(a) Small topology

Stream	Source	Destination	Period (ms)	*Payload (Byte)*	*#Frames*	*Type*
S_1	DA-Cam	HU	1000	46	1	AVB Class A
S_2	DA-Cam	HU	200			
S_3	DA-Cam	CU	1000			
S_4	DA-Cam	CU	200			
S_5	HU	CU	5			
S_6	HU	CU	50			
S_7-S_8	HU	CU	100			
S_9	HU	CU	200			
S_{10}	HU	CU	500			
S_{11}- S_{12}	HU	CU	1000			
S_{13}	HU	DA-Cam	100			
S_{14}- S_{15}	HU	DA-Cam	200			
S_{16}	CU	HU	100	46	25	
S_{17}	CU	HU	200			
S_{18}- S_{19}	CU	HU	500			
S_{20}	CU	HU	1000			
S_{21}	CU	DA-Cam	10			
S_{22}	CU	DA-Cam	1000			
S_{23}- S_{26}	Cam[1-4]	DA-Cam	16.66	128	16	
S_{27}	DA-Cam	HU	16.66			
S_{28}	Telematics	RSE	0.625	120	5	
S_{29}	CD/DVD	RSE	0.25	80	1	

(b) Large Topology

Stream	Source	Destination	Period (ms)	*Payload (Byte)*	*#Frames*	*Type*
S_1	Lidar	VCC1	100	1500	1	AVB Class A
S_2	Cam1	VCC1	100	490		
S_3	Radar1	VCC1	1000	42		
S_4	Cam2	VCC1	100	490		
S_5	Radar2	VCC1	1000	42		
S_6-S_7	Cam[3-4]	VCC1	100	490		
S_8	VCC3	Audio1	100	234		
S_9	VCC3	Audio2				
S_{10}	VCC3	Audio3				
S_{11}	Video	VCC3	100	1500		
S_{12}	VCC2	WinAct1	100	42		
S_{13}	VCC2	WinAct2				
S_{14}	VCC2	WinAct3				
S_{15}	VCC1	BrakeAct1	100	42	1	TT
S_{16}	VCC1	BrakeAct2				
S_{17}	VCC1	BrakeAct3				

Inspired by [12,13,14], Table 3a shows the stream properties used in our experiments for the in-vehicle network in Figure 3a with normal operational conditions subject to no attack. As shown in the table, a total of 29 AVB Class A streams, namely S_1 to S_{29}, are transmitted among the specified nodes, whose period, number of frames per period, and payload range from 0.25 to 1000 ms, 1 to 25, and 46 to 128 bytes, respectively. The streams of S_1 to S_{22} carry the small-sized network control messages associated with ADAS while S_{23} to S_{27} transmit the video traffic. The remaining streams in Table 3a are related to the cd-dvd and telematics data. The period values for the AVB streams specified in Table 3a are taken from [12,13,14], which are determined realistically according to the frequency of the messages generated by the respective in-vehicle applications. Different from [12], the class of S_{29} is selected to be the same as the type of the remaining streams, namely A, since the analysis tool from [17] used in this paper has the limitation that it is not capable of evaluating the interference among AVB streams with a mixture of A and B classes. The deadline of each AVB stream is assumed to be 2 ms in accordance with 802.1BA standard [59] while the capacity of each full-duplex link is selected to be 1 Gbps in both directions.

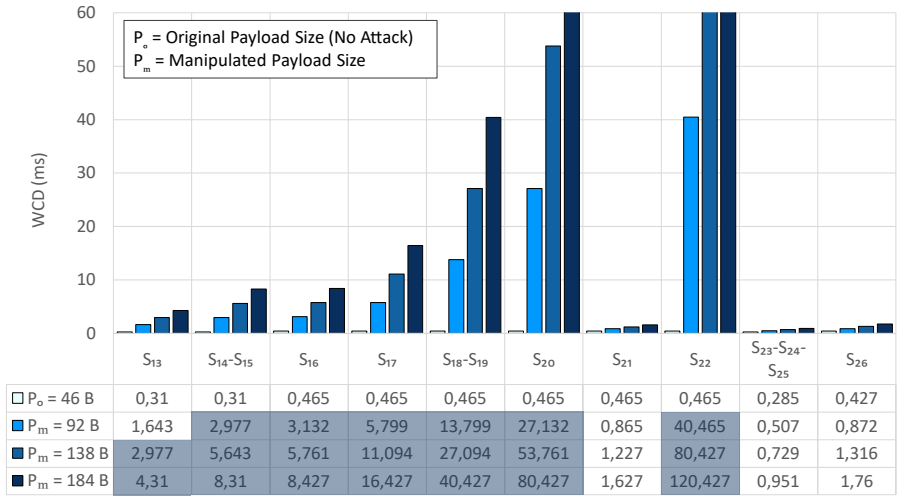
Figure 3b shows the large in-vehicle network topology from a major motor company in China [15] used in our experiments. It contains 8 SWs and 23 ESs distributed over three zones including front left, front right and rear of the vehicle with their own zonal controllers, namely VCC1, VCC2 and VCC3. In order to realize complex in-vehicle communication scenarios, each zone with a star topology implements *Chassis*, *ADAS*, *Infotainment* and *Body* functionalities, and is connected to the other zones via a backbone network with a ring topology composed of SW1, SW2 and SW3 as shown in Figure 3b. The brakes are controlled by the motors in the *Chassis* domain by exchanging control messages with a high priority. The *ADAS* domain contains *Lidar*, *Radar*, camera and controller devices, where lidar, video, millimeter-wave and ultrasonic radar data are communicated, and corresponding control messages are generated. The infotainment domain is used to play video and 3D stereo surround music by transmitting video and audio data messages whereas the body domain contains motors used to control the doors and windows of the vehicle.

Inspired by [15], Table 3b shows the stream properties used in our experiments for the large in-vehicle network in Figure 3b with normal operational conditions subject to no attack. As shown in the table, a total of 14 AVB Class A (S_1 to S_{14}) and 3 TT (S_{15} to S_{17},) streams are transmitted among the specified nodes, whose period and payload range from 100 to 1000 µs, and 42 to 1500 bytes, respectively. On the other hand, the number of frames per period is fixed to 1 for all the streams. The stream parameters specified for the large topology in Table 3b are mostly taken from [15], which are determined based on a real in-vehicle traffic scenario provided by a major motor company in China, taking into consideration the requirements of real-time applications deployed within a vehicle. The traffic types for the streams of S15 to S17 are selected to be TT with the highest priority since the communication between the zonal controllers and braking actuators is critical for the safety of the vehicle. In accordance with [15], the deadline of each AVB stream is assumed to be 8 ms, which is determined based on the time-criticality level of the real-time applications within the vehicle. On the other hand, the capacity of each full-duplex network link is similarly selected to be 1 Gbps in both directions. Shortest-path routes are used for all the streams specified in Table 3.

6.2. Impact Analysis

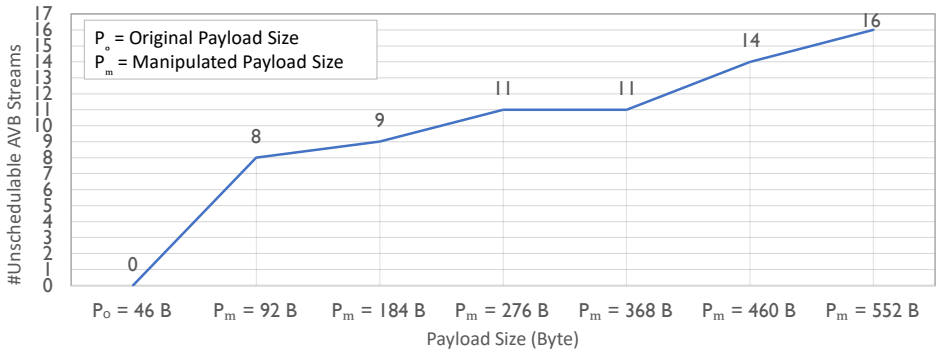
6.2.1. Small-Sized Network

In this section, one of the network components in the small-sized in-vehicle network shown in Figure 3a is assumed to be violated by an attacker to manipulate the original properties shown in Table 3a regarding the AVB streams sourced at the violated unit, and the impact of this manipulation on the delay of other AVB streams transmitted in the network is evaluated. Furthermore, in order to evaluate the impact of an attack scenario targeting the TT traffic, we generate a new TT stream in the network sourced at the violated component, and manipulate its message period and the open duration of the corresponding gates throughout its transmission route. Please note that conducting an attack scenario by generating a new TT stream is more complicated



	S_{13}	S_{14}-S_{15}	S_{16}	S_{17}	S_{18}-S_{19}	S_{20}	S_{21}	S_{22}	S_{23}-S_{24}-S_{25}	S_{26}
P_o = 46 B	0,31	0,31	0,465	0,465	0,465	0,465	0,465	0,465	0,285	0,427
P_m = 92 B	1,643	2,977	3,132	5,799	13,799	27,132	0,865	40,465	0,507	0,872
P_m = 138 B	2,977	5,643	5,761	11,094	27,094	53,761	1,227	80,427	0,729	1,316
P_m = 184 B	4,31	8,31	8,427	16,427	40,427	80,427	1,627	120,427	0,951	1,76

(a) WCD values



(b) The number of unschedulable AVB streams

Figure 4. WCD results for the attacks manipulating the payload size for AVB traffic (D = 2 ms)

compared to a scenario where existing AVB streams are manipulated. This is due to the fact that the generation of a new TT stream in a network requires the manipulation of both the routing tables and GCLs within the switches on the transmission route of the newly generated stream.

6.2.1.1. Manipulation of AVB Traffic

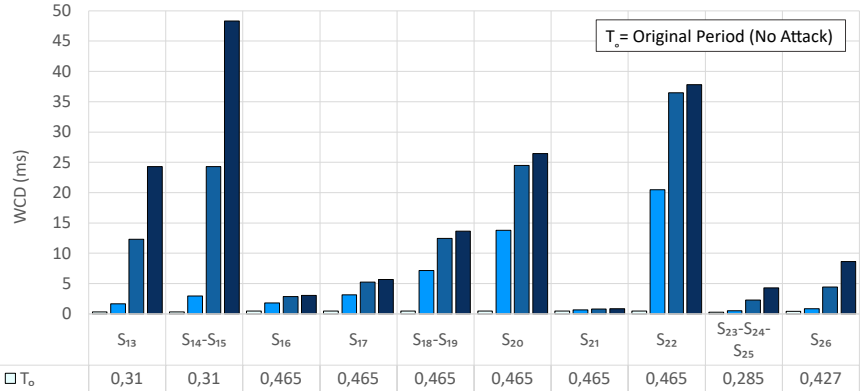
In this section, we assume that the *CU* component shown in Figure 3a is violated by an attacker, who performs a DoS attack by manipulating the properties of a total of 7 AVB streams including S_{16} to S_{22} originating from the violated *CU*. In this section, two different attack scenarios are realized by manipulating either the payload size or message period of the streams sourced at the violated *CU*. As a consequence of these attack scenarios, apart from the manipulated ones, a total of 10 streams including S_1, S_2, S_{13} to S_{15}, S_{23} to S_{27} are impacted since they destine to the same targets, namely *HU* or *DA-Cam*, as the manipulated streams resulting in intersecting communication routes.

Figure 4 shows the experimental results for the attack scenario manipulating the payload size of the frames originating from the *CU*. Figure 4a presents the WCD values in milliseconds for the AVB streams affected from the attack scenario. For this experiment, the deadline of each AVB stream is assumed to be 2 ms in compliance with the 802.1BA standard such that a WCD value exceeding the deadline is specified by a dark background within the accompanying table below the figure. As shown in the figure, P_o represents the original payload size of each frame in the absence of any attack while P_m indicates the payload size determined by the attacker. For this experiment, P_m is selected to be 92, 138 or 184 bytes, while P_o is equal to be 46 bytes in accordance with Table 3a. Figure 4a demonstrates that WCD of a stream, whose transmission route shares some common links with the manipulated streams, significantly increases with respect to the manipulated payload size. For example, the WCD values of S_{14} and S_{15} with the same transmission routes exhibit an identical behaviour by varying as 0.31, 2.977, 5.643 and 8.31 ms for P_o=46, P_m=92, P_m=138 and P_m=184 bytes, respectively. On the other hand, the attack scenario has a much larger impact on the delay characteristics of the manipulated streams themselves. For example, the WCD values of the manipulated stream of S_{22} vary as 0.465, 40.465, 80.427 and 120.427 ms with respect to the payload size. Figure 4b shows the number of unschedulable AVB streams not satisfying the deadline constraints resulting from the attack scenario manipulating the payload size of the frames originating from the *CU*. For this experiment, the number of unschedulable AVB streams is computed using Equation 1. Please note that the number of unschedulable streams cannot exceed 17 for this experiment, which is equal to the total number of streams affected from the attack scenario. Figure 4b shows that the number of unschedulable streams considerably increases with respect to the manipulated payload size. For example, as P_m ranges from 92 to 552 bytes, the respective numbers of unschedulable streams vary as 8, 9, 11, 11, 14 and 16 while all the streams are schedulable for P_o=46 bytes. For this experiment, we also observe that the maximum link utilizations resulting from the attack scenarios never exceed 1.47% due to the majority of the streams in the network transmitting small-sized messages as shown in Table 3a. Please note that we manipulate solely the payload size to realize the attack scenario in Figure 4, and omit the results from manipulating the number of frames per period since increasing the number of frames or payload size at the same rate equally influences the delay characteristics of the impacted streams.

Figure 5 shows the experimental results for an attack scenario manipulating the message period of the streams originating from the violated *CU*, and presents the WCD values in milliseconds for the significantly impacted AVB streams. In Figure 5, T_o represents the original period of each stream in the absence of an attack, while it is reduced by the rates of 50%, 90% or 95% as part of the attack scenario. Figure 5 demonstrates that WCD of a stream, whose transmission route shares common links with the manipulated streams, significantly increases with respect to the manipulated period. For example, the WCD values of S_{14} and S_{15} with the same transmission routes exhibit an identical behaviour by varying as 0.31, 2.977, 24.31 and 48.31 ms for T_o, $T_o \times 0.5$, $T_o \times 0.1$ and $T_o \times 0.05$, respectively. The number of unschedulable AVB streams caused by the attack scenario relying on period reduction shows an increasing trend similar to Figure 4b, and, hence, are omitted from this paper due to space considerations. Similar to the experiments in Figure 4, the maximum link utilizations resulting from the attack scenarios are negligibly small.

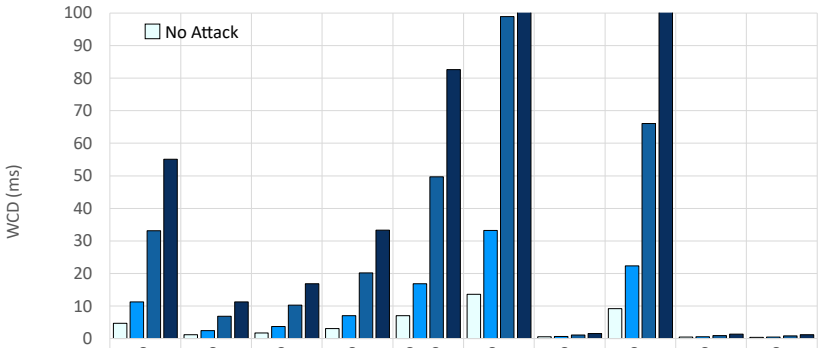
6.2.1.2 Manipulation of TT Traffic

In this section, we assume that the *CU* component shown in Figure 3a is similarly violated by an attacker, who initiates a new TT stream, namely S_{30}, sourced at the violated *CU* and destined to *HU* with the transmission route $[CU, SW_2]$-$[SW_2, SW_1]$-$[SW_1, HU]$. This addition results in a total of 30 real-time streams, namely 29



	S_{13}	S_{14}-S_{15}	S_{16}	S_{17}	S_{18}-S_{19}	S_{20}	S_{21}	S_{22}	S_{23}-S_{24}-S_{25}	S_{26}
T_o	0,31	0,31	0,465	0,465	0,465	0,465	0,465	0,465	0,285	0,427
T_o x 0.5	1,643	2,977	1,799	3,132	7,132	13,799	0,665	20,465	0,507	0,872
T_o x 0.1	12,31	24,31	2,865	5,265	12,465	24,465	0,825	36,465	2,284	4,426
T_o x 0.05	24,31	48,31	3,065	5,665	13,645	26,465	0,839	37,799	4,283	8,646

Figure 5. WCD results for the attacks manipulating the message period for AVB traffic (D = 2 ms)



	S_1	S_2	S_{16}	S_{17}	S_{18}-S_{19}	S_{20}	S_{21}	S_{22}	S_{26}	S_{27}
Duration = 0.04 µs	4,68	1,184	1,777	3,088	7,02	13,575	0,553	9,205	0,5	0,358
Duration = 0.1 µs	11,238	2,496	3,744	7,022	16,858	33,249	0,684	22,321	0,61	0,467
Duration = 0.3 µs	33,124	6,873	10,31	20,155	49,688	98,908	1,122	66,094	0,975	0,832
Duration = 0.5 µs	55,052	11,259	16,89	33,312	82,58	164,69	1,562	109,949	1,34	1,198

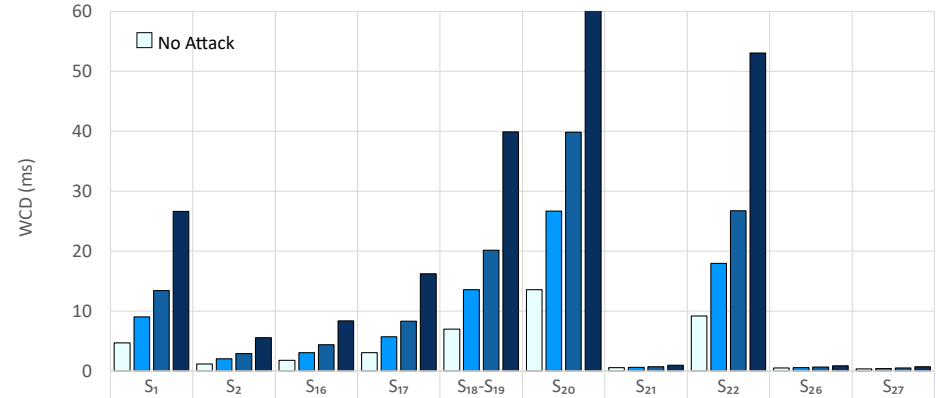
Figure 6. WCD results for the attacks manipulating the open window duration for TT gates (D = 2 ms)

AVB streams shown in Table 3a plus the new S_{30}. Since the TT traffic class has the highest priority in a TSN network, it has a significant impact on the delay of lower-priority traffic especially in a network with a poor path diversity such as the small in-vehicle network shown in Figure 3a.

In this experiment, the attacker varies the severity of his DoS attack by manipulating the properties of S_{30} with varying degrees including the message period and the time duration during which the corresponding gates for the TT traffic are kept open. For this purpose, the open window of each gate located in front of the TT queue in each switch throughout the transmission route of S_{30} is increased so that the lower-priority AVB traffic sharing common links with the TT traffic experiences a higher end-to-end transmission delay. Please note that, within each switch on the transmission route of a TT traffic, the gate in front of the AVB queue is kept closed as long as the gate for the TT traffic is open. Therefore, the AVB traffic suffers a higher level of interference from the TT traffic as the open window for the TT gate increases. As a consequence of these attack scenarios, S_{16} to S_{22} sourced at the violated *CU*, the streams of S_1, S_2 and S_{27} destined to *HU*, and S_{26} are impacted since their transmission routes share common links with S_{30}.

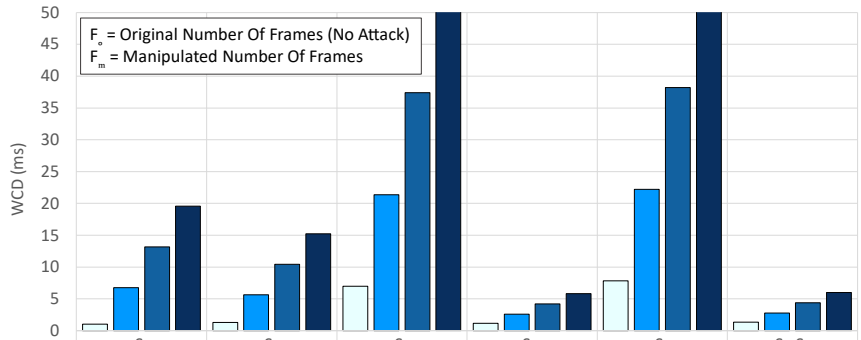
Figure 6 shows the experimental results for the attack scenario manipulating the time duration during which the corresponding TT gates on the transmission route of S_{30} are kept open. For this experiment, the deadline of each AVB stream is similarly assumed to be 2 ms in compliance with the 802.1BA standard. Figure 6 presents the WCD values in milliseconds for the AVB streams affected from the attack scenario, where the open duration ranges from 0.04 to 0.5 µs. Figure 6 demonstrates that WCD of a stream, whose transmission route shares common links with S_{30}, significantly increases with respect to the manipulated duration. For example, the WCD values of S_1 vary as 4.68, 11.238, 33.124 and 55.052 ms for the durations of 0.04, 0.1, 0.3 and 0.5 µs, respectively.

Figure 7 shows the experimental results for an attack scenario manipulating the message period of S_{30}, and presents the WCD values in milliseconds for the impacted AVB streams. As shown in the figure, the message period is varied between 25 and 500 µs. Figure 7 demonstrates that WCD of a stream sharing common links with S_{30} significantly increases with respect to the manipulated period. For example, the WCD values of S_1 vary as 4.68, 9.055, 13.442 and 26.623 ms for T=500, T=125, T=50 and T=25 µs, respectively.



	S_1	S_2	S_{16}	S_{17}	S_{18}-S_{19}	S_{20}	S_{21}	S_{22}	S_{26}	S_{27}
T = 500 µs	4,68	1,184	1,777	3,088	7,02	13,575	0,553	9,205	0,5	0,358
T = 125 µs	9,055	2,059	3,089	5,713	13,583	26,7	0,641	17,955	0,573	0,431
T = 50 µs	13,442	2,937	4,406	8,345	20,164	39,861	0,729	26,729	0,647	0,504
T = 25 µs	26,623	5,573	8,361	16,255	39,936	79,405	0,993	53,092	0,866	0,724

Figure 7. WCD results for the attacks manipulating the period for TT traffic (D = 2 ms)



	S_1	S_2	S_3	S_4	S_5	S_6-S_7
F_o = 1	1,016	1,313	6,981	1,157	7,816	1,342
F_m = 10	6,776	5,633	21,381	2,597	22,216	2,782
F_m = 20	13,176	10,433	37,381	4,197	38,216	4,382
F_m = 30	19,576	15,233	53,381	5,797	54,216	5,982

Figure 8. WCD results for the attacks manipulating the number of frames for AVB traffic (D = 8 ms)

6.2.2. Large-Sized Network

In this section, the *Lidar* component in the large-sized in-vehicle network topology shown in Figure 3b is assumed to be violated. The attacker manipulates the original properties of S_1 shown in Table 3b, which is sourced at the *Lidar*. As a consequence of the realized attack scenario, a total of *7* AVB streams with sensor data including S_2 to S_7 plus the manipulated S_1 are impacted since they destine to the same target, namely *VCC1*, as the manipulated S_1 resulting in intersecting communication routes. Figure 8 shows the experimental results for the attack scenario manipulating the number of frames per period (F) for S_1 originating from the *Lidar.* For this experiment, the deadline of each AVB stream is assumed to be 8 ms inspired by [15]. As shown in the figure, F_0 represents the number of frames per period in the absence of any attack while F_m indicates the number of frames determined by the attacker. For this experiment, F_m is varied as 10, 20 and 30, while F_0 is equal to 1 in accordance with Table 3b.

Figure 8 demonstrates that the WCD of a stream sharing common links with S_1 significantly increases with respect to the manipulated number of frames. For example, the WCD values of S_3 vary as 6.981, 21.381, 37.381 and 53.381 ms for $F_0 = 1$, $F_m = 10$, $F_m = 20$, $F_m = 30$, respectively. We also perform experiments for an attack scenario manipulating the message period of S_1 originating from the violated *Lidar*, where, similar to Figure 8, WCD has an increasing trend with respect to the manipulated period. However, due to space considerations, we omit the results from the paper. For this experiment, we observe that the maximum link utilizations resulting from the attack scenarios exceed the link capacity in all the cases whereas it is only 27% for $F_0 = 1$.

7. Discussion and Future Work

The distributed nature of sensors, actuators and controllers within a CPS such as in-vehicle platforms increases the attack surfaces, making CPS environments more fragile in terms of security. Therefore, the deployment of appropriate defense strategies in in-vehicle platforms plays a crucial role to face all the security issues in in-vehicle embedded systems. For this purpose, an in-vehicle network should be combined with anomaly detection techniques, which can be implemented by using either statistical or machine learning based approaches [60]. Assuming that network traffic follows a certain pattern, statistical techniques for anomaly detection are usually designed by building a probability density function to represent the normal and abnormal network behaviors, and classifying the transmitted traffic relying on this function [61]. On the other hand, machine learning based anomaly detection approaches rely on an initial training phase, where the features of the incoming real-time traffic are extracted by additionally accounting for the clock skew, and then classified into abnormal or normal network behaviors. Following the training phase, the ongoing traffic is classified accordingly based on the learning model. Please note that statistical techniques may provide a lower computational cost compared to the machine learning based detection approaches.

As a successfully launched DoS attack in an in-vehicle platform prevents the timely delivery of control messages originating from controllers, resilient mechanisms for controlling the physical processes within a vehicle tolerating the late or lost control data to a certain degree may be designed as another defense strategy against cyber-attacks. This way, the adverse effects of cyber-attacks on the control quality can be minimized. However, the development of control strategies resilient to cyber-attacks is not within the scope of our research. Other leading countermeasures to maintain security in in-vehicle platforms include i) clustering of electronic components with safety-critical functionalities in security islands, separated from the rest of the network by gateways with reliable cybersecurity functionalities, ii) deployment of hardware accelerators in computing units to perform real-time message encryption, iii) development of novel signature mechanisms for message authentication in computing units, and iv) implementation of gateway firewalls ensuring that only the authorized nodes are able to send valid messages into in-vehicle network [43].

The most suitable location in an in-vehicle platform to host the intrusion detection system is the gateway which has a global knowledge of the transmitted traffic. Therefore, designing TSN-compliant in-vehicle network architectures based on the SDN paradigm with a global view of the underlying network is an ongoing research effort, which may better satisfy the security requirements. However, multiple DoS attacks can be simultaneously launched against sensor-to-controller and controller-to-actuator communication channels within a vehicle. The timely detection of such concurrent attacks may impose a large computational overhead on the central gateway in SDN, which may be even more overwhelming in the case that the launched attacks possess different characteristics for each channel. Such a security problem may only be coped with by designing distributed architectures for the control layer in SDN in order to balance the computational cost among multiple gateways.

The computational cost for implementing the defense strategies in an in-vehicle network should be kept low and upper-bounded in order not to violate the deterministic nature of the underlying real-time communication. In our future work, ensuring an upper-bounded computational cost, we plan to develop statistical and machine learning based anomaly detection approaches along with their respective prevention mechanisms for SDN-based in-vehicle communication networks with a TSN support. For this purpose, in addition to DoS attacks, we consider studying the impacts of other attack types including replay, spoofing, malware and falsified-information on the performance of in-vehicle networks [45]. Finally, we intend to extend the existing open-source Omnet++ simulation models in the literature for network switches and end-systems with a TSN support in order to evaluate the performance of our methodologies for in-vehicle networking [62].

8. Conclusions

In this paper, we present a thorough review of the research work on the security of in-vehicle communication networks with TSN support, and define various DoS attack scenarios targeting the real-time traffic in in-vehicle networks. In order to realize our attack scenarios, the properties of compromised real-time streams including message frequency, payload size and scheduling tables are manipulated. We evaluate the impact of our attack scenarios on the performance of two different realistic in-vehicle communication networks with varying sizes, namely small and large. For the purpose of network delay analysis, we use the AVB Latency Math tool from IEEE 802.1BA standard. The experimental results demonstrate that WCD of victim AVB streams, whose transmission routes share some common links with the manipulated streams, significantly increases with respect to the manipulated stream properties in both small and large experimented in-vehicle networks. Taking into account the considerably limited amount of research work on in-vehicle embedded systems with TSN support, we believe that our paper significantly contributes to the literature by providing an extensive review of the existing work and an impact analysis of DoS attacks targeting in-vehicle platforms. As future work, we plan to investigate the statistical and machine learning based anomaly detection techniques for SDN-based secure in-vehicle networks with TSN support.

Ethics committee approval and conflict of interest statement

This article does not require ethics committee approval.

This article has no conflicts of interest with any individual or institution.

References

[1] Duo, W., Zhou, M., Abusorrah, A. 2022. A Survey of Cyber Attacks on Cyber Physical Systems: Recent Advances and Challenges. IEEE/CAA Journal of Automatica Sinica, 9(5):784-800.

[2] Zhou, Z., Lee, J., Berger, M. S., Park, S. and Yan, Y. 2021. Simulating TSN traffic scheduling and shaping for future automotive Ethernet," in Journal of Communications and Networks, vol. 23, no. 1, pp. 53-62, 2021, doi: 10.23919/JCN.2021.000001.

[3] Neumann, P. 2007. Communication in industrial automation—what is going on? Control Engineering Practice, 15(11):1332–1347. Special Issue on Manufacturing Plant Control: Challenges and Issues.

[4] Aliwa, E., Rana, O., Perera, C. and Burnap, P., 2021. Cyberattacks and countermeasures for in-vehicle networks. *ACM Comput. Surv.*, 54(1).

[5] Thing, V. L. L. and Wu, J. 2016. "Autonomous Vehicle Security: A Taxonomy of Attacks and Defences," 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Chengdu, China, pp. 164-170, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2016.52.

[6] Ashjaei, M., Bello L. L., Daneshtalab, M., Patti, G., Saponara, S., and Mubeen, S. 2021. Time-sensitive networking in automotive embedded systems: State of the art and research opportunities. *Journal of Systems Architecture*, 117:102137.

[7] *IEEE P802.1DG, TSN Profile for Automotive In-Vehicle Ethernet Communications*, 2021.

[8] Pop, P., Raagaard, M. L. 2017. Optimization algorithms for the scheduling of ieee 802.1 time-sensitive networking (tsn). Technical report, Tech. Univ. Denmark.

[9] Pop, P., Raagaard, M. L., Craciunas, S. and Steiner, W. 2016. Design optimization of cyber-physical distributed systems using ieee time-sensitive networks (tsn). *IET Cyber-Physical Systems: Theory & Applications*.

[10] Petit, J. and Shladover, S.E. 2015. Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):546–556.

[11] Waraich, P. S. and Batra, N. 2017. Prevention of denial of service attack over vehicle ad hoc networks using quick response table. In *2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, pages 586–591.

[12] Patti, G. and Bello, L.L. 2019. Performance Assessment of the IEEE 802.1Q in Automotive Applications. In 2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), pp. 1-6.

[13] Leonardi, L., Bello, L. L. and Patti, G. 2020. Performance assessment of the IEEE 802.1Qch in an automotive scenario. In 2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), pp. 1-6.

[14] L., Bello, Ashjaei, M., Patti, G. and Behnam, M. 2020. Schedulability analysis of time-sensitive networks with scheduled traffic and preemption support. *Journal of Parallel and Distributed Computing*, 144:153– 171.

[15] Luo, F., Wang, B., Yang, Z., Zhang, P., Ma, Y., Fang, Z., Wu, M. and Sun, Z. 2022. Design methodology of automotive time-sensitive network system based on omnet++ simulation system. *Sensors*, 22(12).

[16] Zhao, L., Pop, P. and Craciunas, S.S. 2018. Worst-Case Latency Analysis for IEEE 802.1Qbv Time Sensitive Networks Using Network Calculus. IEEE Access, 6:41803-41815.

[17] Laursen, S.M., Pop, P. and Steiner, W. 2016. Routing optimization of avb streams in tsn networks. *SIGBED Rev.*, 13(4):43–48.

[18] Topsakal, M. and Cevher, S. 2022. Impact Analysis of Denial of Service Attacks in IEEE 802.1 Time Sensitive Networking. In 30th IEEE Signal Processing and Communications Applications Conference (SIU), pp. 1-4.

[19] Navet N. and Simonot-Lion, F. 2013. In-vehicle communication networks - a historical perspective and review. Industrial Communication Technology Handbook, Second Edition, CRC Press Taylor&Francis.

[20] Alves, R. 2008. A glimpse into the future of travel and its impact on marketing, in: IEEE-SA Ethernet and IP at Automotive Technology day (EIPATD). https://standards.ieee.org/wp-content/uploads/import/documents/other/eipatd-presentations/2019/D2-01_ALVES-Design_and_Implementation_of_IDS_for_AVB-TSN_Networks.pdf. [Online; accessed January-2023].

[21] Ergenc, D., Brulhart, C., Neumann, J., Kruger, L., Fischer, M.. 2021. On the Security of IEEE 802.1 Time-Sensitive Networking," 2021 IEEE International Conference on Communications Workshops, Montreal, QC, Canada, pp. 1-6, doi: 10.1109/ICCWorkshops50388.2021.9473542.

[22] Bello L.L. and Steiner W. 2019. A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems, in Proceedings of the IEEE, vol. 107, no. 6, pp. 1094-1120, doi: 10.1109/JPROC.2019.2905334.

[23] Mahfouzi, R., Aminifar, A., Samii, S., Eles, P. and Peng, Z. 2019. Security-aware Routing and Scheduling for Control Applications on Ethernet TSN Networks. *ACM Trans. Des. Autom. Electron. Syst.* https://doi.org/10.1145/3358604.

[24] Reusch, N., Craciunas, S.S. and Pop, P. 2022. Dependability-aware routing and scheduling for time-sensitive networking. *IET Cyber-Physical Systems: Theory & Applications*, 7(3):124–146.

[25] Wüsteney, L., Menth, M., Hummen, R. and Heer, T. 2021. Impact of packet filtering on time-sensitive networking traffic. In *2021 17th IEEE International Conference on Factory Communication Systems (WFCS)*, pages 59–66.

[26] Pena, R.A., Pascual, M., Astarloa, A., Uribe, D. and Inchausti, J. 2022. Impact of macsec security on tsn traffic. In 2022 *37th Conference on Design of Circuits and Integrated Circuits (DCIS)*, pages 01–06.

[27] Tang, S., Hu, X. and Zhao, L. 2020. Modeling and security analysis of ieee 802.1as using hierarchical colored petri nets. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, doi: 10.1109/GLOBECOM42002.2020.9347988.

[28] Kobzan, T., Schriegel, S., Althoff, S., Boschmann, A., Otto, J. and Jasperneite, J. 2018. Secure and time-sensitive communication for remote process control and monitoring. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 1105– 1108.

[29] Li, H., Li, D., Zhang, X., Shou, G., Hu, Y. and Liu, Y. 2021. A security management architecture for time synchronization towards high precision networks. *IEEE Access*, 9:117542–117553.

[30] Böhm, M., Ohms, J., Gebauer, O. and Wermser, D. 2018. Architectural design of a tsn to sdn gateway in the context of industry 4.0.

[31] Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F. and Kohno, T. 2011. Comprehensive experimental analyses of automotive attack surfaces. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, page 6, USA, USENIX Association.

[32] Koscher, K. et al. Experimental Security Analysis of a Modern Automobile. 2010. 2010 IEEE Symposium on Security and Privacy, Oakland, CA, USA, pp. 447-462, doi: 10.1109/SP.2010.34.

[33] Miller, C., Valasek, C. 2013. Adventures in Automotive Networks and Control Units.

[34] Woo, S., Jo H. J. and Lee, D. H. 2015. A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN. in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 993-1006, doi: 10.1109/TITS.2014.2351612.

[35] Miller, C., Valasek, C. 2014. A survey of remote automotive attack surfaces.

[36] Miller, C., Valasek, C. 2015. Remote Exploitation of an Unaltered Passenger Vehicle. Black Hat USA 2015.

[37] Chen, Y. et al. Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-driving Vehicle. 2016.

[38] Zeng, K., Liu, S., Shu, Y., Wang, D., Li, H., Dou, Y., Wang, G. and Yang, Y. 2018. All your GPS are belong to us: towards stealthy manipulation of road navigation systems. In Proceedings of the 27th USENIX Conference on Security Symposium (SEC'18). USENIX Association, USA, 1527–1544.

[39] Nie S. and Liu, L. 2017. Free-fall: Hacking tesla from wireless to can bus. Technical report, Keen Security Lab of Tencent.

[40] Nie S., Liu, L., Du, Y., Zhang, W. 2018. Over-the-air: how we remotely compromised the gateway and autopilot ECUs of Tesla cars, Keen Security Lab of Tencent, Black Hat USA.

[41] Cai, W.Z.Z. and Wang, A. 2019. 0-days & mitigations: Roadways to exploit and secure connected bmw cars. Technical report, Keen Security Lab of Tencent.

[42] Mercedes benz mbux security research report. 2021. Technical report, Keen Security Lab of Tencent.

[43] Bello, L. L., Mariani, R., Mubeen, S. and Saponara, S. 2019. Recent advances and trends in on-board embedded and networked automotive systems. *IEEE Transactions on Industrial Informatics*, 15(2):1038–1051.

[44] Lin, C.W. and Yu, H. 2016. Invited: Cooperation or competition? coexistence of safety and security in next-generation ethernet-based automotive networks. In *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6.

[45] Häckel, T., Meyer, P., Korf F. and Schmidt, T. C. 2023. Secure Time-Sensitive Software-Defined Networking in Vehicles. IEEE Transactions on Vehicular Technology, vol. 72, no. 1, pp. 35-51, doi: 10.1109/TVT.2022.3202368.

[46] Meyer, P., Häckel, T., Korf F. and Schmidt, T. C. 2020. Network Anomaly Detection in Cars based on Time-Sensitive Ingress Control," *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, Victoria, BC, Canada, pp. 1-5, doi: 10.1109/VTC2020-Fall49728.2020.9348746.

[47] Luo, F., Wang, B., Fang, Z., Yang, Z., Jiang, Y. and Demertzis, K. 2021. Security Analysis of the TSN Backbone Architecture and Anomaly Detection System Design Based on IEEE 802.1Qci. Sec. and Commun.Netw.2021. https://doi.org/10.1155/2021/6902138

[48] Meyer, P. 2016. Preventing dos attacks in time sensitive networking in-car networks through credit based ingress metering.

[49] Aoudi, W., Nowdehi, N., Almgren, M. and Olovsson, T. 2021. Spectra: detecting attacks on in-vehicle networks through spectral analysis of CAN-message payloads. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing* (SAC '21). Association for Computing Machinery, New York, NY, USA,1588–1597. https://doi.org/10.1145/3412841.3442032.

[50] Bozdal, M., Samie, M. and Jennions, I. K. 2021. WINDS: A Wavelet-Based Intrusion Detection System for Controller Area Network (CAN). IEEE Access, 9, 58621-58633.

[51] Nowdehi, N., Aoudi, W., Almgren, M., Olovsson, T. 2019. CASAD: CAN-Aware Stealthy-Attack Detection for In-Vehicle Networks.

[52] Han, M. L., Kwak B. I. and Kim H. K. 2021. Event-Triggered Interval-Based Anomaly Detection and Attack Identification Methods for an In-Vehicle Network. IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2941-2956, doi: 10.1109/TIFS.2021.3069171.

[53] Cho K. T. and Shin, K.G. 2016. Fingerprinting electronic control units for vehicle intrusion detection. *In Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, page 911–927, USA. USENIX Association.

[54] Waszecki, P., Mundhenk, P., Steinhorst, S., Lukasiewycz, M., Karri, R. and Chakraborty, S. 2017. Automotive electrical and electronic architecture security via distributed in-vehicle traffic monitoring. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(11):1790–1803.

[55] Specht, J. and Samii, S. 2017. Synthesis of queue and priority assignment for asynchronous traffic shaping in switched ethernet. In *2017 IEEE Real-Time Systems Symposium (RTSS)*, pages 178–187.

[56] Demir, Ö.K. and Cevher, S. 2023. Multi-Topology Routing based traffic optimization for IEEE 802.1 Time Sensitive Networking. Real-Time Syst, 59:123–159.

[57] Alshammari, A., Zohdy, M., Debnath, D. and Corser, G. 2018. Classification approach for intrusion detection in vehicle systems. *Wireless Engineering and Technology*, 09:79–94.

[58] Ji, H., Wang, Y., Qin, H., Wu, X. and Yu, G. 2018. Investigating the effects of attack detection for in-vehicle networks based on clock drift of ecus. *IEEE Access*, 6:49375–49384.

[59] IEEE Standard for Local and Metropolitan Area Networks--Audio Video Bridging (AVB) Systems. 2021. IEEE Std 802.1BA-2021 (Revision of IEEE Std 802.1BA-2011).

[60] Bhuyan, M., Bhattacharyya, D. K. and Kalita, J. 2017. Network Traffic Anomaly Detection and Prevention: Concepts, Techniques, and Tools.

[61] Markou, M. and Singh, S. 2003. Novelty detection: a review - part 1: statistical approaches. *Signal Process.*, 83:2481–2497.

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