engineering proceedings

Proceeding Paper

Delay-Compensated Time-Aware Shaper for Automotive Time-Sensitive Networks†

Sol-Bee Lee [1], Inhyeok Kang [2], Jung-Hyok Kwon [1] and Eui-Jik Kim [3,*]

[1] Smart Computing Laboratory, Hallym University, 1 Hallymdaehak-gil, Chuncheon 24252, Republic of Korea; thfqla3535@hallym.ac.kr; jhkwon@hallym.ac.kr
[2] Department of Convergence Software, Hallym University, 1 Hallymdaehak-gil, Chuncheon 24252, Republic of Korea; M23522@hallym.ac.kr
[3] Division of Software, Hallym University, 1 Hallymdaehak-gil, Chuncheon 24252, Republic of Korea; ejkim32@hallym.ac.kr
* Correspondence: ejkim32@hallym.ac.kr
† Presented at the title, place, and date.

Abstract: This paper proposes a delay-compensated time-aware shaper (DC-TAS) designed to mitigate cascading delays in automotive time-sensitive networks (TSNs). DC-TAS dynamically adjusts the time windows allocated to scheduled traffic (ST), non-scheduled traffic (NST), and guard bands (GB) based on the available idle time within the NST and GB windows. This adaptive approach effectively compensates for delays in both ST and best-effort traffic (BE) caused by unexpected emergency traffic (ET) surges, while maintaining the deterministic transmission guarantees essential for safety-critical applications. Simulation results demonstrate that DC-TAS outperforms the enhanced TAS (eTAS), achieving lower end-to-end delay and higher throughput. This makes DC-TAS a robust solution for complex and unpredictable in-vehicle network (IVN) environments.

Keywords: cascading delays; IEEE 802.1Qbv; in-vehicle network (IVN); time-aware shaper (TAS); time-sensitive networking (TSN); time window adaptation

Academic Editor: Firstname Lastname

Published: date

Citation: To be added by editorial staff during production.

1. Introduction

As in-vehicle networks (IVNs) become increasingly complex and require higher reliability and lower latency, there is a growing need for advanced networking solutions that can meet these stringent demands. To address this, time-sensitive networking (TSN) is gaining prominence as a key technology for next-generation IVNs. TSN, proposed by the IEEE 802.1 TSN task group, is a set of standards designed to enhance Ethernet by providing real-time capabilities, deterministic transmission, time synchronization, frame preemption, and redundancy [1]. IEEE 802.1Qbv (time-aware shaper, TAS) is especially essential in IVNs as it facilitates deterministic scheduling for time-critical and safety-critical traffic [2]. Furthermore, TAS ensures predictable data transmission by prioritizing various types of traffic, making it highly suitable for IVNs where diverse traffic types associated with multiple services and functions coexist [3].

In IVNs, traffic can be classified into three main types based on their characteristics and transmission requirements: scheduled traffic (ST), best-effort traffic (BE), and emergency traffic (ET). ST is primarily used for safety-critical data and real-time control signals, requiring strict transmission guarantees such as low latency, deterministic delivery, and low jitter. BE is generally used for non-time-critical data transmission and does not require any specific constraints. ET is unpredictable and aperiodic, potentially involving accident detection and system safety events, thereby necessitating reliable transmission. TAS provides a mechanism to control the transmission of traffic using a gate control list (GCL), which operates by opening or closing specific gates for each timeslot. However, TAS does not specify how frames of different traffic classes are scheduled. Furthermore, TAS does not provide a robust solution to address delays that ST may experience due to unexpected events such as switching and processing delays or sudden surges of ET in complex and unpredictable operating environments.

Several prior studies have investigated TAS scheduling issues with a focus on ET handling. Kim *et* al. [4] proposed enhanced TAS (eTAS), a dynamic scheduling scheme that prioritizes real-time transmission of emergency ETs while minimizing disruptions to prescheduled STs. eTAS modifies the queuing rules of legacy TAS to prioritize emergency ETs as the highest priority and introduces a time window extension algorithm that temporarily extends the time window for transmissions of ST. Through this approach, eTAS mitigates cascading delays caused by the transmission of emergency ETs. However, if an unexpectedly large number of ETs arrive at the queue and cannot be fully accommodated through time window extension, eTAS may still fail to prevent cascading delays.

In this paper, we propose a delay-compensated TAS (DC-TAS) for automotive TSNs. DC-TAS operates by introducing a time window adaptation (TWA) algorithm that dynamically updates the GCL in response to ET arrivals. Upon ET arrival, the TWA algorithm calculates the required transmission time and adjusts the start time and duration of the current ST, NST, or guard band (GB) time windows. If ET arrives during the ST window, the window is extended by the ET transmission time. If ET arrives during the NST window, it is extended by taking into account both the ET transmission time and any available idle time in the NST window. The GCL is updated accordingly. To verify the superiority of DC-TAS, we conducted simulation experiments. The results show that DC-TAS outperforms eTAS by an average of 8.91% in end-to-end delay and 32.04% in throughput, respectively.

2. Overview of Time-Aware Shaper (TAS)

TAS was introduced for TSN to meet the stringent requirements of time-critical traffic, such as low and deterministic latency and low-jitter. TAS employs multiple queues and timed-gate mechanisms to control each traffic class independently, which helps to isolate time-critical frames from potential interference and delays caused by other traffic classes. TAS includes eight queues, each corresponding to a specific traffic class, with priorities from 7 (highest) to 0 (lowest). When a frame arrives at the switch, its priority is identified based on the priority code point (PCP) value in the IEEE 802.1Q Ethernet header [5]. The frame is then assigned to the queue corresponding to its priority level. Each queue is connected to a gate, which can be in either an open (o) or closed (C) state. A frame in a queue can only be transmitted when the corresponding gate is open. These gates are independently controlled by the GCL. If multiple gates are open simultaneously and more than one queue has pending frames, the transmission selection process selects and transmits the frame with the highest priority. The GCL entries are repeated cyclically over a preconfigured time period. In addition, TAS operates based on a cyclic time schedule consisting of three types of time windows: ST, NST, and GB. ST time window is allocated for prescheduled time-critical traffic. NST time window is used for BE and all other traffic. GB time window serves to prevent interference with traffic scheduled for transmission in the upcoming ST time window. To this end, all gates remain closed during the GB time window. The GB time window always precedes the ST time window. The length of the GB window is set to be sufficient for transmitting the largest Ethernet frame.

3. Design of DC-TAS

DC-TAS is designed to minimize the impact of ET on ST and NST by mitigating the cascading delays caused by surges in ET. To achieve this, DC-TAS provides a time window adaptation (TWA) algorithm. We assume that ET has the highest priority among all traffic types and its gate for ET is always open. In the following, we provide further details of DC-TAS.

Algorithm 1 presents the procedure for the TWA, which is triggered when ET arrives at the switch. The algorithm takes l_{ET} and $\mathbf{GCL}$ as inputs. l_{ET} denotes the size of the arriving ET. $\mathbf{GCL}$ refers to the GCL of the switch. The output of the algorithm is the updated GCL, denoted as $\mathbf{GCL}'$.

The algorithm first calculates the transmission time of the ET (t_{ET}). It then checks the current time window ($currentWindow$). If the current time window is for NST, it calculates its idle time (t_{CW}^{idle}). The idle time refers to the duration during which no frames are transmitted according to the GCL. Then, if either the idle time or the combined duration of the idle time and the GB time window (t_{GB}) exceeds the transmission time required for the ET frame, it shifts the start time of the next time window (t_{next}) by the transmission time of the ET. As a result, the NST time window is extended, and the GB time window is correspondingly reduced. Otherwise, if the current time window is for ST, the start time of the next time window is also shifted by the same amount. Meanwhile, if the current window is for GB and its duration exceeds the transmission time of the ET, the GCL remains unchanged. Finally, the algorithm returns the updated GCL ($\mathbf{GCL}'$).

Algorithm 1. Time Window Adaptation

Input: l_{ET}, $\mathbf{GCL}$
Output: $\mathbf{GCL}'$
Procedure:

$t_{ET} \leftarrow (l_{ET} \cdot 8)/R$
if $currentWindow == NST$
  $t_{CW}^{idle} \leftarrow getIdleTime(currentWindow)$
  if $(t_{CW}^{idle} > t_{ET}) \,||\, (t_{CW}^{idle} + t_{GB} > t_{ET})$
    $t_{next} \leftarrow t_{next} + t_{ET}$
  end if
else if $currentWindow == ST$
  $t_{next} \leftarrow t_{next} + t_{ET}$
else
  if $t_{GB} < t_{ET}$
    $\mathbf{GCL}' \leftarrow \mathbf{GCL}$
  end if
end if
return $\mathbf{GCL}'$

4. Performance Evaluation

We conducted experimental simulations under two scenarios to evaluate the performance of DC-TAS using the OMNeT++ 6.1 simulator [6] with INET 4.5 framework [7]. The simulation results were compared with eTAS [4]. In the following subsections, we provide a detailed description of the simulation configuration and simulation results.

4.1. Simulation Configuration

In the simulation, the IVN consists of 11 transmitters, 9 receivers, and 4 switches. All sensors, actuators, and devices are connected to each other by a 1-meter, 100-Mbps Ethernet links. Figure 1 depicts the network topology of the IVN. To evaluate the performance of DC-TAS under different traffic conditions, we consider two ET scenarios where an ET is generated randomly every 100μs and every 500μs (i.e., 100μs scenario and 500μs scenario), respectively. Table I shows the traffic configuration for our simulations. Each traffic flow is assigned fixed values for priority, traffic class, interval, and payload size. The cycle time of the GCL is set to 500μs. Specifically, ST, NST, and GB durations are set to 123.04μs, 106.72μs, 270.24μs, respectively.

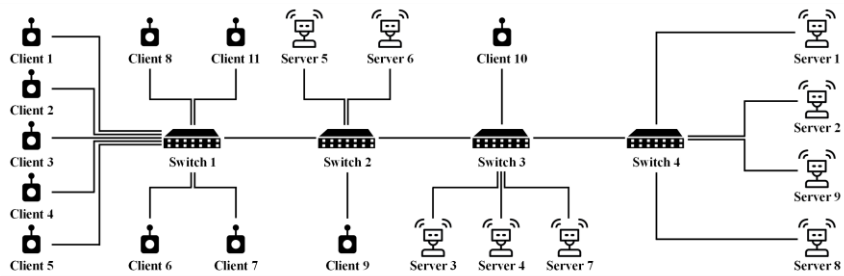


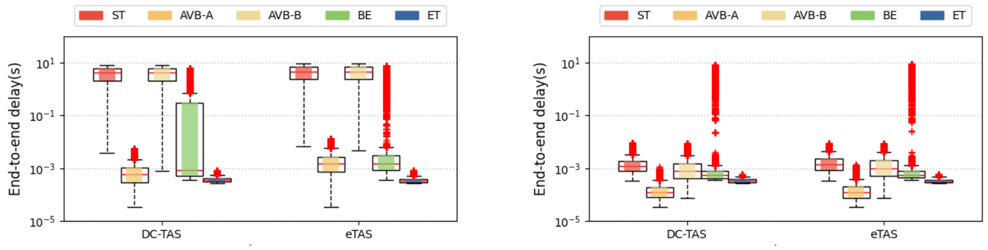
Figure 1. Network Topology of the IVN.

Table 1. Traffic Configuration.

Traffic Type	Traffic Class	Source Node	Destination Node	Interval (μs)	Payload Size (Byte)
ST	4	Client 1	Server 1	500	625
		Client 2	Server 2	500	625
AVB-A	6	Client 3	Server 3	125	46
		Client 4	Server 4	125	46
AVB-B	5	Client 5	Server 5	250	250
		Client 6	Server 6	250	256
		Client 7	Server 7	250	256
BE	0	Client 8	Server 8	550	1400
		Client 9		675	1400
		Client 10		646	1400
ET	7	Client 11	Server 9	Random	625

4.2. Simulation Results

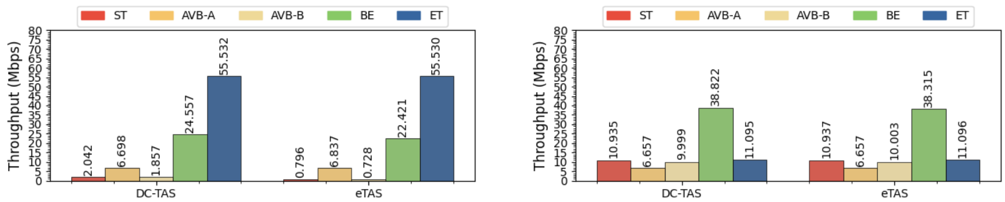
Figures 2(a) and 2(b) show the end-to-end delay distributions of DC-TAS and eTAS under 100μs and 500μs scenarios, respectively, with results presented on a logarithmic scale. The results indicate that DC-TAS effectively mitigates prolonged queuing delays, which may occur in eTAS due to the frequent arrival of ET traffic. Specifically, DC-TAS demonstrated on average 14.55%, 33.79%, and 16.53% lower end-to-end delays for ST, AVB-A, and AVB-B, respectively. This latency reduction is attributed to DC-TAS's dynamic adjustment of both ST and NST time windows via the TWA algorithm, which shortens queuing times for traffic. Moreover, ST, AVB-A, and AVB-B that would otherwise be deferred to the next cycle due to the sudden arrival of ET traffic can be processed within the current cycle through dynamic window extension, thereby reducing overall waiting time. As a result, DC-TAS minimizes the cascading delay caused by the prioritized processing of ET. Quantitatively, DC-TAS achieved on average 12.78% and 5.05% lower end-to-end delays than to eTAS in the 100μs and 500μs scenarios, respectively.



(**a**) (**b**)

Figure 2. End-to-end delay (s): (a) 100µs and (b) 500µs scenarios.

Figures 3(a) and 3(b) show the throughput performance of DC-TAS and eTAS under 100µs and 500µs scenarios, respectively. The results indicate that DC-TAS achieves either higher or comparable throughput across all traffic types relative to eTAS. Specifically, in the 100µs scenario, DC-TAS achieved throughput improvements of 156.36%, 155.24%, and 9.53% for ST, AVB-B, and BE, respectively, compared to eTAS. This throughput enhancement results from the dynamic extension of both ST and NST time windows in DC-TAS, which increases transmission opportunities for traffic relative to eTAS. In contrast, the ET throughput of DC-TAS and eTAS shows similar performance, as both schemes maintain a dedicated ET queue that remains always open to preserve the highest priority for ET. Meanwhile, in the 500µs scenario, the throughput for all traffic types is nearly identical. This is because, despite the frequent occurrence of ET traffic, both schemes are capable of sufficiently accommodating ET through dynamic adjustment of time windows. Quantitatively, DC-TAS achieved on average of 63.82% higher throughput than eTAS in the 100µs scenario, demonstrating significant improvements in network utilization efficiency through the proposed TWA algorithm.



(**a**) (**b**)

Figure 3. Throughput (Mbps): (a) 100µs and (b) 500µs scenarios.

5. Conclusion

In this paper, we propose DC-TAS for automotive TSNs to mitigate the cascading delays. DC-TAS employs a flexible time window adaptation mechanism that modifies the duration assigned to ST and NST according to the unused capacity detected in NST and GB time windows. This strategy successfully mitigates transmission delays for ST, AVB-A, and AVB-B traffic resulting from unpredictable ET, while preserving the deterministic scheduling requirements crucial for mission-critical systems. To demonstrate the superiority of DC-TAS, we conducted experimental simulations. The results show that DC-TAS not only improves individual traffic class performance but also enhances overall network efficiency. By minimizing cascading delay and optimizing time window utilization, DC-TAS reduces network congestion and improves resource utilization efficiency. Quantitatively, DC-TAS outperforms eTAS by 8.91% and 32.04% on average in terms of end-to-end delay and throughput, respectively.

Author Contributions: Conceptualization: S.-B.L., I.K., J.-H.K. and E.-J.K.; methodology: S.-B.L., I.K., J.-H.K. and E.-J.K.; software: S.-B.L., I.K., J.-H.K. and E.-J.K.; validation: S.-B.L. and E.-J.K.; formal analysis: S.-B.L. and E.-J.K.; investigation: J.-H.K. and E.-J.K.; resources: S.-B.L. and E.-J.K.; data curation: I.K. and J.-H.K.; writing—original draft preparation: S.-B.L., I.K., J.-H.K. and E.-J.K.; writing—review and editing: S.-B.L., I.K., J.-H.K. and E.-J.K.; visualization: S.-B.L.; supervision: E.-J.K.; project administration: E.-J.K.; funding acquisition: E.-J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00353365). This work was supported by the NRF grant funded by the Korea government (MSIT) (RS-2021-NR062134).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

BE	Best-effort Traffic
DC-TAS	Delay-Compensated Time-Aware Shaper
ET	Emergency Traffic
eTAS	Enhanced TAS
GB	Guard Band
GCL	Gate Control List
IVN	In-Vehicle Network
NST	Non-Scheduled Traffic
PCP	Priority Code Point
ST	Scheduled Traffic
TAS	Time-Aware Shaper
TSN	Time-Sensitive Networking
TWA	Time Window Adaptation

References

1. IEEE Time-Sensitive Networking Task Group. Available online: http://www.ieee802.org/1/pages/tsn.html (accessed on 18 July 2025)
2. IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015); IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment 25: Enhancements for Scheduled Traffic. IEEE: Piscataway, NJ, USA, 2016; pp. 1–57.
3. Chahed, H.; Kassler, A. TSN Network Scheduling—Challenges and Approaches. *Network* **2023**, *3*, 585-624.
4. Kim, M.; Hyeon, D.; Paek, J. eTAS: Enhanced Time-Aware Shaper for Supporting Non-Isochronous Emergency Traffic in Time-Sensitive Networks. *IEEE Internet Things J.* **2021**, *9*, 21860262.
5. IEEE Std 802.1 Q-2018; IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks. (Revision of IEEE Std 802.1Q-2014). IEEE: New York, NY, USA, 2018; pp. 1–1993.
6. OMNeT++ Discrete Event Simulator. Available online: https://omnetpp.org/ (accessed on 18 July 2025)
7. INET Framework. Available online: https://inet.omnetpp.org/ (accessed on 18 July 2025)