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

**Sol-Bee Lee[1], Inhyeok Kang[2], Jung-Hyok Kwon[1], Eui-Jik Kim[3]***
[1]Smart Computing Laboratory, Hallym University, Chuncheon, Republic of Korea
[2]Department of Convergence Software, Hallym University, Chuncheon, Republic of Korea
[3]Division of Software, Hallym University, Chuncheon, Republic of Korea
thfqla3535@hallym.ac.kr, M23522@hallym.ac.kr, jhkwon@hallym.ac.kr, ejkim32@hallym.ac.kr

## Introduction

### Time-aware Sensitive Networking (TSN)

- TSN is gaining prominence as a **key technology for next-generation In-vehicle networks (IVNs)**.
- TSN is a set of standards designed to enhance Ethernet by **providing real-time capabilities, deterministic transmission, time synchronization, frame preemption, and redundancy.**

### Time-Aware Shaper (TAS)

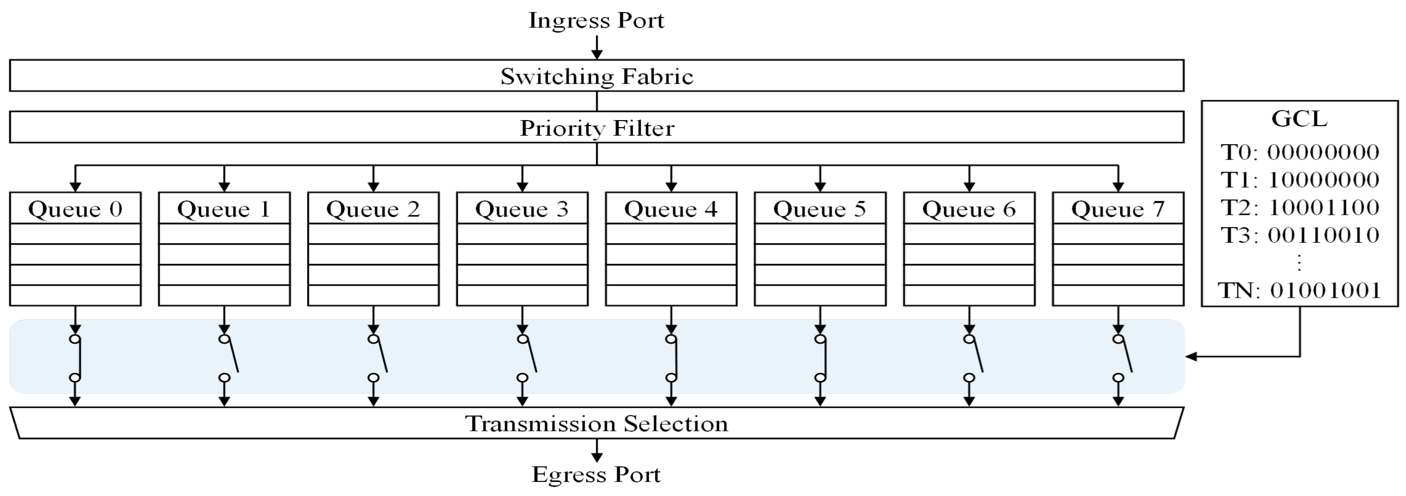
- TAS is especially essential in IVNs as it **facilitates deterministic scheduling for time-critical and safety-critical traffic.**
- TAS **ensures predictable data transmission by prioritizing various traffic types**, and employs **eight queues and timed-gate mechanisms**.
- **Each queue** corresponding to a specific traffic class, with **priorities from 7 to 0**, and it **is connected to a gate**, which can be in either an **open or closed state**.
- These **gates are independently controlled by the gate control list (GCL).** TAS uses **a cyclic time schedule consisting of three types of time windows**.
- **Scheduled traffic (ST)**
  - ✓ ST time window is allocated for prescheduled time-critical traffic.
- **Non-scheduled traffic (NST)**
  - ✓ NST time window is used for BE and all other traffic.
- **Guard band (GB)**
  - ✓ 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.


### Traffic Types in IVNs

- Traffic can be classified into **three types** based on their characteristics and transmission requirements.
- **Scheduled traffic (ST)**
  - ✓ 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.
- **Best-effort traffic (BE)**
  - ✓ BE is used for non-time-critical data transmission and does not require any specific constraints.
- **Emergency traffic (ET)**
  - ✓ ET is unpredictable and aperiodic, potentially involving accident detection and system safety events, thereby necessitating reliable transmission.

### Motivation

- **TAS does not specify how frames of different traffic classes are scheduled.**
- **Furthermore, TAS does not provide a robust solution to address delays that scheduled traffic may experience due to unexpected events.**




## DC-TAS: Delay-compensated TAS

### 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.**
- We assume that **ET has the highest priority** among all traffic types and **its gate for ET is always open.**
- DC-TAS operates by introducing a **time window adaptation (TWA) algorithm that dynamically updates the GCL in response to ET arrivals.**

### Operation of TWA

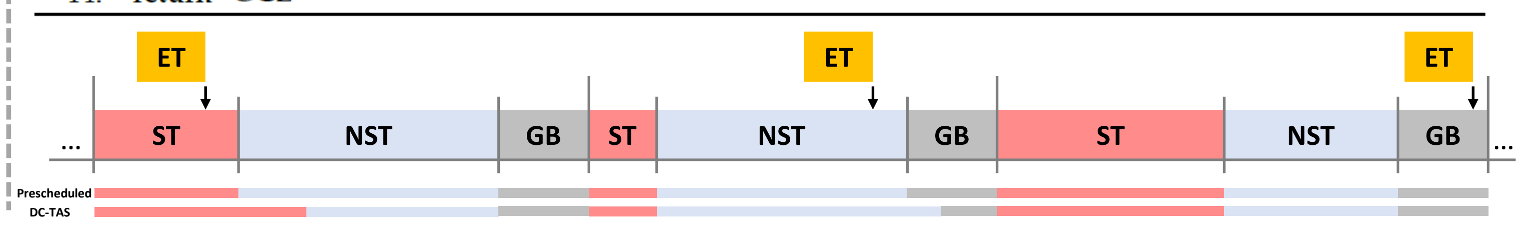
1) Calculate the transmission time of the ET and check the current time window.
2) If the current time window is for NST, calculate its idle time.
3) If either the idle time or the combined duration of the idle time and the GB time window exceeds the transmission time required for the ET frame, shift the start time of the next time window by the transmission time of the ET.
4) If the current time window is for ST, shift the start time of the next time window by the ET transmission time.
5) If the current time window is for GB and its duration exceeds the ET transmission time, keep the GCL unchanged.
6) Return the updated GCL.

### Time Window Adaptation Algorithm

**Algorithm 1.** Time Window Adaptation

**Input:** $l_{ET}$, GCL
**Output:** GCL′
**Procedure:**

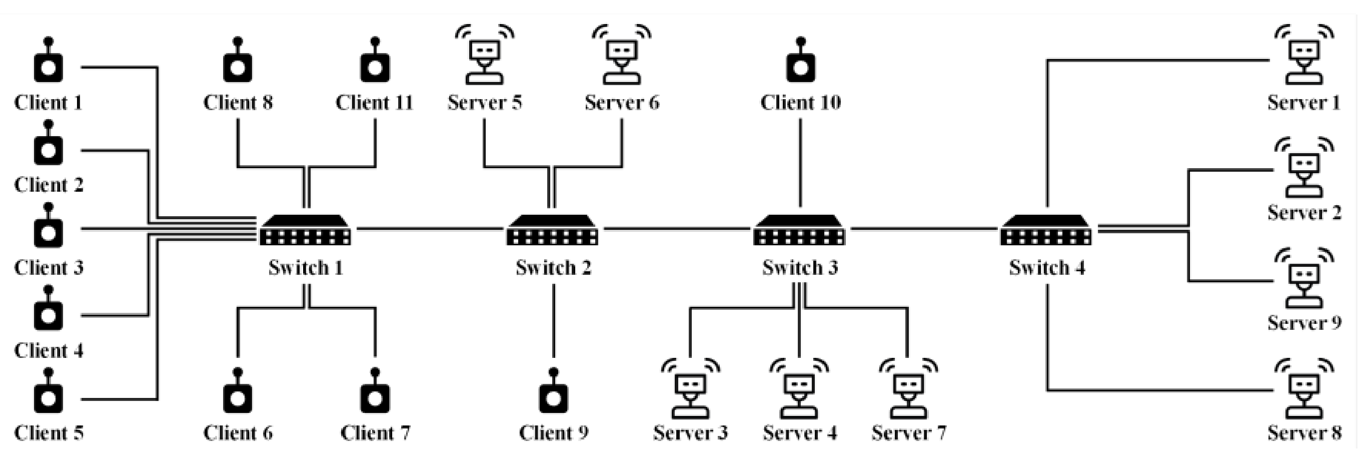
$t_{ET} \leftarrow (l_{ET} \cdot 8) / R$
**if** $currentWindow == NST$
  $t_{CW}^{idle} \leftarrow getIdleTime(currentWindow)$
  **if** $(t_{CW}^{idle} > t_{ET}) \| (t_{CW}^{idle} + t_{GB} > t_{ET})$
    $t_{next} \leftarrow t_{next} + t_{ET}$
  **end if**
**else if** $currentWindow == ST$
  $t_{next}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




## Performance Evaluation

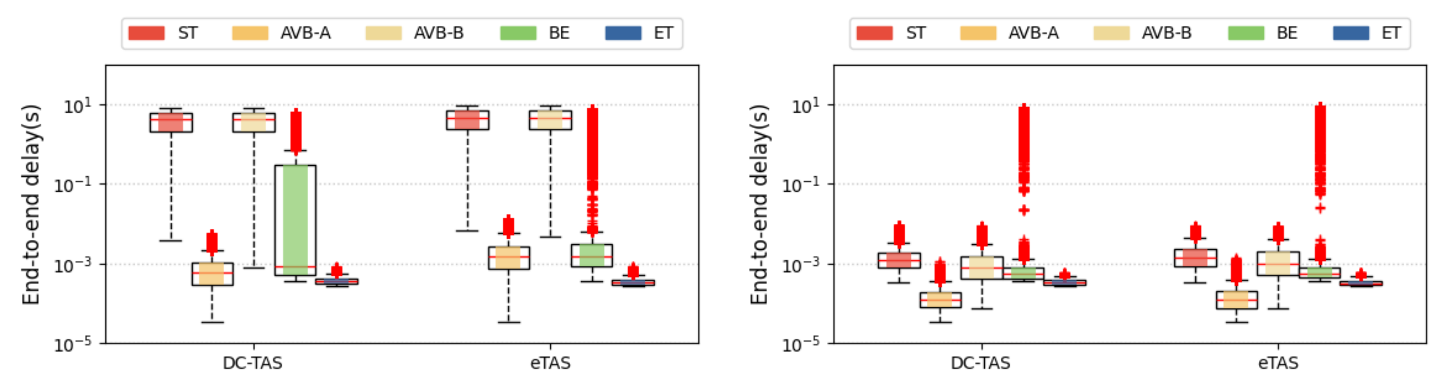
### Simulation Configuration

- We **conducted experimental simulations under two scenarios** to evaluate the performance of DC-TAS **using the OMNeT++ 6.1 simulator with INET 4.5 framework.**
- 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.
- We consider two ET scenarios where an ET is generated randomly every 100μs and every 500μs (i.e., 100μs scenario and 500μs scenario), respectively.
- The cycle time of the GCL is set to 500μs.



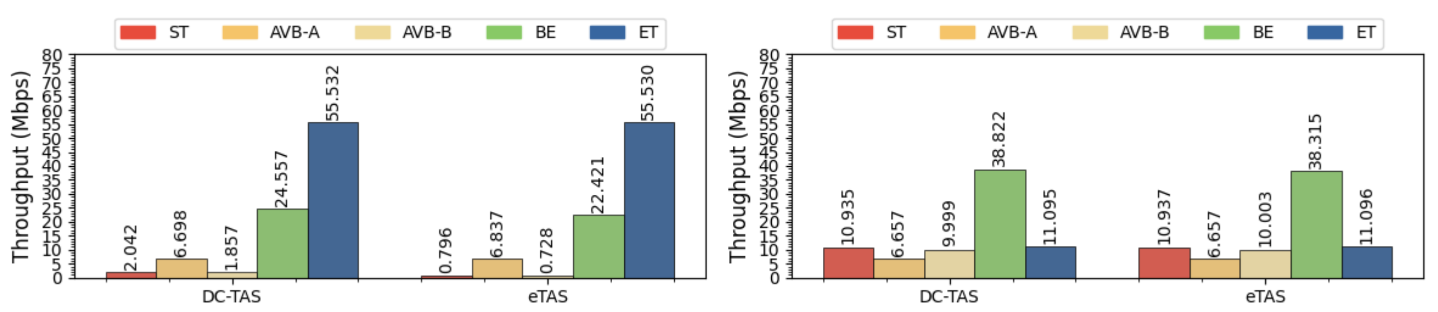

### Simulation Results

- 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) 100μs scenario (b) 500μs scenario

- DC-TAS achieved **on average of 63.82% higher throughput** than eTAS in the 100μs scenario.




(a) 100μs scenario (b) 500μs scenario

ICKII 8th International Conference on Knowledge Innovation and Invention 2025 (ICKII 2025) Paper ID: K250221-H