Quantitative Performance Comparison of Various Traffic Shapers in Time-Sensitive Networking

Luxi Zhao[a], Paul Pop[b], and Sebastian Steinhorst[a]

[a]Department of Electrical and Computer Engineering, Technical University of Munich, Germany
[b]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark

***Abstract*—Owning to the sub-standards being developed by IEEE Time-Sensitive Networking (TSN) Task Group, the traditional IEEE 802.1 Ethernet is enhanced to support real-time dependable communications for future time- and safety-critical applications. Several sub-standards have been recently proposed that introduce various traffic shapers (e.g., Time-Aware Shaper (TAS), Asynchronous Traffic Shaper (ATS), Credit-Based Shaper (CBS), Strict Priority (SP)) for flow control mechanisms of queuing and scheduling, targeting different application requirements. These shapers can be used in isolation or in combination and there is limited work that analyzes, evaluates and compares their performance, which makes it challenging for end-users to choose the right combination for their applications. This paper aims at (i) quantitatively comparing various traffic shapers and their combinations, (ii) summarizing, classifying and extending the architectures of individual and combined traffic shapers and their Network calculus (NC)-based performance analysis methods and (iii) filling the gap in the timing analysis research on handling two novel hybrid architectures of combined traffic shapers, i.e., TAS+ATS+SP and TAS+ATS+CBS. A large number of experiments, using both synthetic and realistic test cases, are carried out for quantitative performance comparisons of various individual and combined traffic shapers, from the perspective of upper bounds of delay, backlog and jitter. To the best of our knowledge, we are the first to quantitatively compare the performance of the main traffic shapers in TSN. The paper aims at supporting the researchers and practitioners in the selection of suitable TSN sub-protocols for their use cases.**

I. Introduction

NOWADAYS, modern cyber-physical and embedded systems, including systems in the automotive, industrial automation, avionics and aerospace domain increasingly depend on the real-time capabilities of their communication networks. Time-Sensitive Networking (TSN) [1] enhances standard Ethernet [2], aiming at providing deterministic communication for real-time traffic. Over the recent years, TSN has become a high-profile and active standardization effort with a strong research community both in academia and in the industry. TSN integrates multiple traffic types implemented by different scheduling mechanisms (traffic shapers), such as the Time-Aware Shaper (TAS) standardized by IEEE 802.1Qbv [4], the Asynchronous Traffic Shaper (ATS) standardized by IEEE 802.1Qcr [5], the Credit-Based Shaper (CBS) standardized by IEEE 802.1Qav [7]. These shapers can be used separately or in several combinations. TAS is based on a global clock synchronization (via IEEE 802.1AS [8]) implementing the time-triggered traffic to guarantee deterministic transmission. ATS avoids using the global clock synchronization, but it is still able to provide real-time guarantees by reshaping traffic flows per hop to reduce the burstiness of traffic. CBS is an asynchronous traffic shaper that implements a bandwidth reservation mechanism.

Many related works have already been proposed for the schedulability analysis and configuration for different traffic shapers. For TAS, which relies on global clock synchronization, the scheduling synthesis for time-triggered (TT) traffic, which is also called as scheduled traffic (ST), has been studied in [11]–[15] using different implementation methods to synthesize Gate Control Lists (GCLs). Vlk et al. [15] increase the schedulability and throughput of scheduled traffic (ST) by proposing a simple hardware enhancement of a switch. Ramon et al. [16] relaxes the constraints of the scheduling model to increase the solution space at the expense of the deterministic scheduling of TAS. A more flexible class-based (i.e., window-based) TAS model is proposed in [17], [18], which does not require strict flow isolation in queues and supports unscheduled end systems. Reusch et al. [39] propose the class-based schedule synthesis for 802.1Qbv. Craciunas et al. [10] gives an overview of the comparison of scheduling mechanisms for TAS in TSN networks and time-triggered scheduling in TTEthernet. In [20], researchers solved the stability-aware integrated scheduling and routing problem for networked cyber-physical systems based on the 802.1Qbv TSN standard. ATS is developed from the urgency-based scheduler (UBS) proposed by Specht et al. [21] and aims at achieving low latency without designing time schedules harmonized among all end systems and switches based on global time synchronization. The same authors [22] propose the synthesis of queues and priority assignment for ATS. Zhou et al. [23], [24] present the simulation model of ATS implemented in Riverbed simulator. [25] proves that ATS will not introduce extra overheads to the worst-case delay of the FIFO system. For CBS, several performance and schedulability analysis methods have been proposed in [26]–[30].

The previous studies all assume a single traffic shaper used individually. There are also some limited studies on the combination of different traffic shapers. An overview of the combined usage of TAS and CBS in controlling flows in in-vehicle networks was presented in [31]. A simulation study of the coexistence of TAS and CBS is presented in [32]. Zhao et al. [33] propose the performance analysis of AVB traffic under the coexistence of CBS and TAS. The same authors [34] extend the timing analysis for arbitrary number of AVB classes under the same architecture of TAS+CBS, considering both standard credit behavior and more generally assumed credit behavior but deviating from the standard 802.1Q [1]. Mohammadpour et al. [35] consider the combination of non-time-triggered control-data traffic (CDT), CBS and ATS, and give the latency and backlog bounds for the traffic of CBS affected by ATS. However, the CDT model is not a standard model required by the TSN standard. In [36], researchers present a simulation

E-mail: luxi.zhao@tum.de; paupo@dtu.dk; sebastian.steinhorst@tum.de.

model of combined CBS and ATS within the OMNeT++ simulator.

With the increasing number of sub-standards for TSN networks, there have been several literature reviews related to TSN networks. Researchers [37] have given a comprehensive survey on TSN networks, from TSN sub-standards to the existing research of TSN before 2018. Maile et al. [38] provide an overview of the exsiting publications that use a Network Calculus approach in the timing analysis for TSN networks. Researchers in [19] make a comparison between flow-based (i.e., frame-based) and class-based TAS, which concludes that class-based scheduling is easy to plan but loses the advantages of extremely low latency and jitter compared with the flow-based TAS. Nasrallah et al. [40] presented the performance comparison of class-based TAS and ATS based on simulations. Nevertheless, the ATS architecture they considered does not exactly match the general model of ATS [5], [21]. They apply the ATS shaper at the ingress port of the switch instead, and consider another extra urgent queue with the highest priority before ST traffic. Thus, as stated above, there are currently no comprehensive and systematic guidelines on quantitative performance comparison of different traffic shapers, and their further coexistence possibilities and interactions in TSN networks.

This paper aims at (i) quantitatively comparing various traffic shapers, i.e., TAS, ATS, CBS, strict priority (SP) scheduling and their combinations, (ii) summarizing, classifying and extending the architectures of individual and combined traffic shapers and their performance analysis methods and (iii) filling the gap in the timing analysis research handling on these novel combinations. We consider the coexistence between time-triggered shapers (TAS) and various event-triggered shapers (ATS, CBS, SP). Our findings will support the researchers and practitioners in understanding the performance characteristics and mutual effect of different traffic shapers. The main contributions of the paper are as follows,

- We summarize the architectures of the main traffic shapers and their combinations in TSN. In order to perform a fair comparison, we use the same method (Network Calculus, NC) to evaluate the performance of each shaper. Based on ours and other researchers' existing NC-based analysis work for different traffic shapers in TSN, we summarize, classify and extend them. For example, inspired by [35], we summarize NC-based analysis for the ATS shaper used individually, and compare in the experiment with the closed-form formula proposed by [21]. We complete the general uniform formula for timing analysis of arbitrary number of AVB classes when CBS is used individually.
- Two novel hybrid architectures of traffic shapers, i.e., TAS+ATS+SP (compared with TAS+SP) and TAS+ATS+CBS (compared with TAS+CBS) are proposed to understand the impact of ATS reshaping on the combined architectures. The NC-based timing analysis method is extended to analyze the real-time performance of traffic in these combinations. The combinations have been selected to provide a comprehensive coverage of possible combined traffic shapers in TSN networks, supported by their corresponding NC-based performance analysis.
- A large number of experiments, using both synthetic and realistic test cases, for quantitative performance comparisons of various individual and combined traffic shapers are carried out, from the perspective of upper bounds of delay, backlog and jitter. Especially with ATS shaping, we highlight interesting results that do not always show the superiority of ATS compared with other shapers, in isolation or combination. We aim at providing a basic reference for the selection of the suitable TSN sub-protocols for researchers and practitioners.

TABLE I
SUMMARY OF NOTATION.

Symbol	Meaning
f	A flow
l_f	Frame size of the flow f
T_f	Period for periodic flow f
P_f	Priority of the flow f
b_f	Burst of the leaky bucket model of the flow f
r_f	Rate of the leaky bucket model of the flow f
R_f	Route of the flow f
D_{pro}	Propagation delay
D_{fwd}	Forwarding delay in the switch
$d_Q(t)$	Queuing delay of frames in the queue Q
D_Q	Latency upper bound of frames waiting in the queue Q
$D_{Q,f}$	Latency upper bound of flow f waiting in the queue Q
$D_{\text{E2E},f}$	Upper bound of end-to-end latency of the flow f
$d_{\text{E2E},f}$	Lower bound of end-to-end latency of the flow f
B_Q	Backlog upper bound of the queue Q
$J_{\text{E2E},f}$	Upper bound of end-to-end jitter of the flow f
h	Output port of a node (link)
h^-	Output port of a preceding node connected to h
C	Physical link rate
ϕ_f^h	Start transmission time (offset) of the flow f on the link h
n_{SP}	Priority number of SP traffic
n_{CBS}	Class number of AVB traffic
$\alpha_Q(t)$	Input arrival curve of flows arriving before the queue Q
$\beta_Q(t)$	Service curve supplied for flows waiting in the queue Q
$\alpha_Q^*(t)$	Output arrival curve of flows departing from the queue Q
$\sigma^{\text{link}}(t)$	Shaping curve of the physical link
$\sigma^{\text{CBS}}(t)$	Shaping curve of CBS
$\delta_D^q(t)$	Burst-delay function
Q_i	Queue of traffic with priority i in the current node port
Q_i^-	Queue of traffic with priority i in the preceding node port
$l_{>i}^{\max}$	Maximum frame size in traffic with priority lower than the priority i
$l_Q^{\max}$	Maximum frame size in the queue Q
$l_Q^{\min}$	Minimum frame size in the queue Q
$idSl_i$	Idle slope for AVB traffic of Class M_i
$sdSl_i$	Send slope for AVB traffic of Class M_i
$c_i^{\max}$	Credit upper bound for AVB Class M_i (no GB / credit frozen during GB)
$\overline{c}_i^{\max}$	Credit upper bound for AVB Class M_i (credit non-frozen during GB)
$c_i^{\min}$	Credit lower bound for AVB Class M_i
L^{GB}	Maximum guard band duration (μs)
$\wedge$	$x \wedge y = \min\{x, y\}$
$[f(t)]_{\uparrow}^{+}$	$\max_{0 \le s \le t}\{f(s), 0\}$

The remainder of the paper is organized as follows. Sect. II gives the overview of performance metrics for the TSN traffic evaluation. Sect. III summarizes and supplements the worst-case performance analysis for traffic transmission with individual TAS, ATS and CBS shapers. Sect. IV presents novel combined architectures of shapers, and extends the NC-based analysis. The evaluation of our performance comparison of individual traffic shapers and their combinations is provided in Sect. V. Sect. VI concludes of the paper. The background of the NC method used is briefly introduced in Appendix A.

II. Overview of Performance Metrics in TSN Evaluation

In this paper, we will compare the quality of service for each individual and combined traffic shapers from the perspective of upper bounds of end-to-end latency, backlog and end-to-end jitter. The end-to-end latency is the time a frame uses to traverse

the network from the sending node to the receiving node along its route. The latency upper bound is a significant QoS metric for real-time applications, which is used to check if a message meets its deadline. The backlog is defined as the number of bits waiting in the queue to be served at any time, and the backlog upper bound can be used to determine the buffer size needed to avoid frame loss. The jitter represents the variation in the latency of a flow. High amounts of jitter indicate poor network performance.

In this paper, flows manipulated by time-triggered shapers (TAS) can only be periodic flows, and flows handled by the event-triggered shapers (ATS, CBS, SP) can be periodic or aperiodic flows. For a periodic flow, we know its frame size, period and priority, i.e., $\langle l_f, T_f, P_f \rangle$. For a sporadic flow, we assume as the related work that the flow is regulated by a leaky bucket model (b_f, r_f) before entering the network [21], where b_f and r_f are the burst and rate of the leaky bucket, respectively. Thus for a sporadic flow we know its $\langle b_f, r_f, l_f, P_f \rangle$. In this paper, the traffic class (priority) P_f for the flow f remains the same on all nodes along its path. TSN supports at most eight different priorities (0 of lowest - 7 of highest priority). Table I summarizes the notations used in this paper.

2.1. End-to-End Latency Upper Bounds

Considering a flow f, its source of delay consists of: (i) Propagation delay D_{pro}, which is tightly related to the physical medium, and considered as constant in this paper; (ii) Forwarding delay D_{fwd} on the switch, which is the time interval from the time after the frame being fully received, to the time it arrives at the buffer located after the switching fabric. It is also generally considered constant; (iii) Queuing delay $d_Q(t)$ on the egress port, which is a time-variant depending on the flows' contention on the port. The upper bound of queuing delay D_Q can be calculated based on the Network Calculus theory [41], see Appendix A. By constructing the input arrival curve $\alpha_Q(t)$ of aggregate flows before Q, which represents the upper envelope of flows arrival in any time interval, and the service curve $\beta_Q(t)$, which represents the service guarantee for these flows, the upper bound of queuing latency of any flows in the queue Q can be calculated by the maximum horizontal deviation of $\alpha_Q(t)$ and $\beta_Q(t)$,

$$D_{Q,f} = D_Q = \mathrm{h}\left(\alpha_Q(t), \beta_Q(t)\right), \tag{1}$$

which is also the upper bound of delay for each flow f in Q. Then, the upper bound of end-to-end delay of the flow f is obtained by the sum of delays from the source ES to the destination ES along its route R_f,

$$D_{\text{E2E},f} = \sum_{Q \in R_f} \left(D_f^Q + D_{\text{pro}} + D_{\text{fwd}} \right) - D_{\text{fwd}}. \tag{2}$$

2.2. Backlog Upper Bounds

According to the Network Calculus theory, the upper bound of backlog in a queue Q is given by the maximum vertical deviation between the arrival curve $\alpha_Q(t)$ of aggregate flows before the queue Q and the service curve $\beta_Q(t)$ offered to flows waiting in the queue Q,

$$B_Q = \mathrm{v}\left(\alpha_Q(t), \beta_Q(t)\right). \tag{3}$$

2.3. End-to-End Jitter Upper Bounds

Jitter refers to the delay variation, i.e., the difference in end-to-end latency between any selected frames in a flow transmitting over a network. Then, the upper bound of jitter of a flow f is calculated by the difference between the maximum and minimum bounds of end-to-end latency of the flow f. The upper bound of end-to-end latency $D_{\text{E2E},f}$ has been discussed previously in Eq. (2). The lower bound of end-to-end latency $d_{\text{E2E},f}$ of f is the sum of transmission delays along its route without the interference from other flows, which can be given as follows,

$$d_{\text{E2E},f} = \sum_{Q \in R_f} \left(l_f / C + D_{\text{pro}} + D_{\text{fwd}} \right) - D_{\text{fwd}}. \tag{4}$$

Thus the upper bound of jitter for the flow f is,

$$J_{\text{E2E},f} = D_{\text{E2E},f} - d_{\text{E2E},f}. \tag{5}$$

As shown above, in order to obtain the performance metrics for different traffic shapers in TSN, the main objective is to construct the arrival curve $\alpha_Q(t)$ and service curve $\beta_Q(t)$ for the corresponding traffic shapers.

III. Performance Analysis of Individual Traffic Shapers

In the following, the performance analysis for each individual traffic shaper, including Time-Aware Shaper (TAS), Asynchronous Traffic Shaper (ATS) and Credit Based Shaper (CBS), are summarized and extended. Their quantitative performance comparison in Sect. 5.1V-A is based on these analyses. When discussing a certain traffic shaper, it is assumed that all nodes, including end systems (ESes) and switches (SWs), in the network support this traffic shaper.

3.1. Time-Aware-Shaper (TAS)

Relying on the global network clock (IEEE 802.1ASrev [3]), IEEE 802.1Qbv [4] defines the Time Aware Shaper (TAS) used to control a gate for each queue of the output port to enable time-triggered communication, enabling the deterministic transmission of extremely low latency and jitter using Gate Control Lists (GCLs). In this paper, we consider the flow-based TAS [12]–[15], which is a widely used model compared to class-based TAS [17], [18]. Moreover, [19] has concluded that it has much better performance in terms of latency and jitter compared with the class-based TAS.

Fig. 1 depicts a TAS architecture in an egress port of a node supporting 802.1Qbv. The switching fabric forwards input flows to the corresponding output port according to their routing information. The traffic class filtering (TCF) dispatches input frames to the corresponding queue of the output port according to their traffic class. For each egress port, there are eight queues, where there may be multiple queues used for TT traffic as to achieve completely deterministic transmission, depending on the TT traffic load and construction of GCLs. Frames waiting in a queue are eligible for transmission only when the corresponding queue gate is open. The TAS control is implemented based on GCLs which dictate the state of the gates. The open and closed states are represented by 1 and 0 respectively in GCLs, as shown with the GCL table beside the TAS architecture in Fig. 1. For example, at time t_1, the gate for the queue Q_2 is open (1) while all the rest are closed (0). Full control of frames can be implemented by mutually exclusive opening queues.

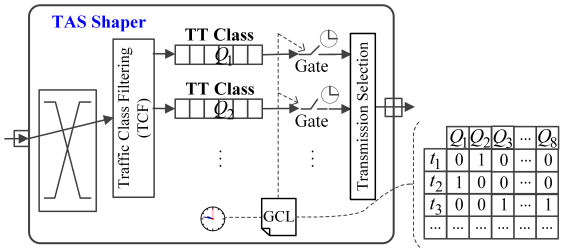


Fig. 1. TAS Architecture

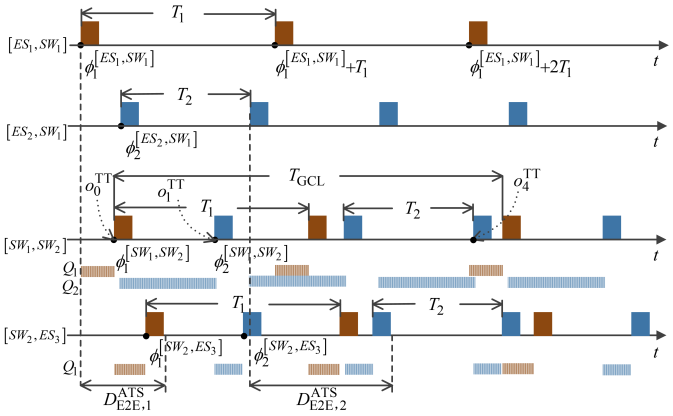


Fig. 2. Example GCL synthesis for ST

Currently, the flow-based TAS can only support periodic traffic scheduling. The problem of GCL synthesis is to find feasible and optimized offsets and queue allocations for periodic traffic. A frame of a TT flow f on a link (egress port) $h = [v_a, v_b]$ is defined by the tuple $\langle \phi_f^h, l_f/C \rangle$, of which ϕ_f^h and l_f/C are respectively denoting the start time (i.e., offset) of transmission and transmission duration of the frame on the respective link. Flow f repeatedly sends frames at times ϕ_f^h, $\phi_f^h + T_f$, $\phi_f^h + 2 \cdot T_f$, $\phi_f^h + 3 \cdot T_f$, ... Fig. 2 shows an example of a GCL using a Gantt chart, describing the transmission of two TT flows f_1 and f_2, with the routes $r_1 = [[ES_1, SW_1], [SW_1, SW_2], [SW_2, ES_3]]$ and $r_2 = [[ES_2, SW_1], [SW_1, SW_2], [SW_2, ES_3]]$, respectively. The x-axis represents the time dimension, and the y-axis lists the output ports. The rectangles represent the TT frames' transmission, of which length equals to l_f/C. The left side of the rectangle is the start time of the transmitted frame, which equals to ϕ_f^h. The thin shaded row labeled Q_i below the link represents the waiting time of the frame in the corresponding queue Q_i.

It can be seen that the transmission time of TT traffic is scheduled in advance. Thus, the performance metrics can be obtained together with GCLs synthesis without the need of complex performance analysis methods.

3.1.1. Performance Analysis – TAS

End-to-End Latency Bounds - TAS. A flow f using flow-based TAS has a completely deterministic end-to-end latency. When the GCL is constructed, its end-to-end delay is known by the time duration between the sending time $\phi_f^{h_0}$ on the source ES h_0 and the reception time $\phi_f^{h_n^-} + l_f/C$ on the destination ES h_n. During the design phase of determining the offset ϕ_f^h, the following inherent delays are considered: propagation delay D_{pro}, forwarding delay D_{fwd}, network precision δ due to the time-synchronization, and store-and-forward (transmission) delay l_f/C, which enforces that a frame is forwarded by a node only after it has been fully received at the node. Then, the end-to-end latency for the TT flow f is given by,

$$D_{\text{E2E},f}^{\text{TAS}} = \phi_f^{h_n^-} + l_f/C - \phi_f^{h_0}. \tag{6}$$

Backlog Bounds - TAS. In order to fully control each frame transmission, researchers have proposed [12], [13] to isolate the frames in queues, i.e., at most one flow occupies a queue at a time, preventing the frame transmission ordering from being disrupted. We depict the queue occupancy with thin shaded rows in Fig. 2. Then, the backlog bounds in the queue Q is the maximum frame assigned to such a queue,

$$B_Q^{\text{TAS}} = \max_{f \in Q} \{l_f\}. \tag{7}$$

Jitter bounds - TAS. Real-time communications are typically sensitive to jitter. The flow-based TAS model [12]–[15] that implements a completely deterministic transmission leads to zero jitter, i.e.,

$$J_{\text{E2E},f}^{\text{TAS}} = 0. \tag{8}$$

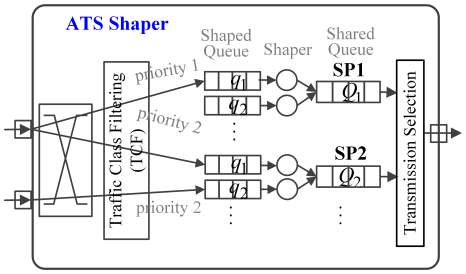


Fig. 3. ATS Architecture

3.2. Asynchronous Traffic Shaper (ATS)

Asynchronous Traffic Shaping (ATS) is another real-time traffic type, standardized by IEEE P802.1Qcr [5]. It uses asynchronous transmission and although it does not require a global clock, it uses the local clocks to reshape traffic in each node. The ATS shaper is associated with the output port, of which the architecture is shown in Fig. 3. It contains two levels of queues, (i) shaped queue q and (ii) shared queue Q, and the ATS shaping algorithm located between them. Shaped queues are used to pre-store frames, which are waiting to be reshaped into the leaky bucket model by the ATS shaping algorithm. Shared queues are used for different priority traffic, and there are at most 8 shared queues. The shared queues follow the strict priority scheduling mechanism. ATS has been proposed with the goal to avoid burstiness cascades. To which shaped queue the frames should enter depends on the queuing schemes as follows,

- QAR1: frames from different senders (input ports) should not be assigned to the same shaped queue in the receiver;
- QAR2: frames from the same sender but with different priority levels are not allowed to be assigned to the same shaped queue;
- QAR3: frames are not allowed to be stored in the same shaped queue if the frames sent to receivers are in different priority levels.

The number of shaped queues in the receiver is related to the number of senders and the number of priorities assigned to the traffic from the sender to the receiver. Since, in the paper, traffic priority for a flow remains the same at all nodes along its path,

the number of shaped queues is only related to the number of senders (used input ports).

The ATS shaping algorithm (Sect. 8.6.11.3 in [5]) is derived from the Token Bucket Emulation (TBE) algorithm, which implements the committed transmission rate r_f and the committed burst size b_f for each flow by calculating an eligible time for frame transmission. Note that ATS is not implemented with per-flow queuing, but the ATS shaping algorithm needs to record state per flow in order to reshape each flow with the respective constraint [25]. It also means that the shaping parameters in the ATS shaping algorithm [5, § 8.6.11.3] are for per-flow but not for per-queue. Frames waiting in each shaped queue are forwarded into the shared queue in FIFO order, following their respective eligible transmission times.

According to the proof in [25] that ATS will not introduce extra overheads to the worst-case delay of the FIFO system, and inspired by [35] of the combined ATS and CBS performance analysis based on NC method, we summarize, for the first time, a NC approach to analyze the performance of ATS used individually, which forms the basis of supporting combinations of ATS with other shapers. Moreover, we compare the NC-based method with the closed-form formula proposed by [21] in the experiment in Sect. 5.1.11.

3.2.1. Performance Analysis – ATS

Service Curve $\beta_{Q_i}^{\text{ATS}}(t)$ - ATS - Shared Queue. The service for the traffic in the shared queues obeys strict priority scheduling, i.e., flows with low priority can obtain the service only when the queues of higher priority traffic are empty. Then, by ATS reshaping, the service curve for SP traffic with priority i ($i \in [1, n_{\text{SP}}]$) in the corresponding shared queue Q_i is given by,

$$\beta_{Q_i}^{\text{ATS}}(t) = C\left[t - \frac{\sum_{j=1}^{i-1} \alpha_{Q_j}^{\text{ATS}}(t)}{C} - \frac{l_{>i}^{\max}}{C}\right]^{+}, \tag{9}$$

where $[x]^{+} = \max\{0, x\}$, $\alpha_{Q_j}^{\text{ATS}}(t)$ (Eq. (11)) is the aggregate arrival curve of SP flows after ATS reshaping with the priority j higher than the priority i, and $l_{>i}^{\max} = \max_{j>i}\{l_{Q_j}^{\max}, l_{Q_{\text{BE}}}^{\max}\}$ that is the maximum frame size of traffic with the priority lower than priority i.

Input Arrival Curve $\alpha_{Q_i}^{\text{ATS}}(t)$ - ATS - Shared Queue. The input arrival curve $\alpha_{Q_i}^{\text{ATS}}(t)$ of aggregate SP flows with priority i before the shared queue Q_i is related to the total output arrival curves $\alpha_q^*(t)$ of individual flows from each previous shaped queue q. As mentioned, each flow is reshaped into the leaky bucket model before entering the shared queue. Then, the output arrival curve of an individual flow f from the shaped queue satisfies $r_f \cdot t + b_f$, where r_f and b_f are the committed transmission rate and burst size for the flow f implemented by ATS shaping algorithm, respectively. Note that, in this paper, if flow f is aperiodic, b_f and r_f are set to the same leaky bucket parameters as before f entered the network, and if f is periodic, we have $b_f = l_f$ and $r_f = l_f/T_f$. The output arrival curve of aggregate flows from the shaped queue q is the sum of output arrival curves of individual flows in q,

$$\alpha_q^*(t) = \sum_{f \in q} \left(r_f \cdot t + b_f\right). \tag{10}$$

Moreover, according to the queuing schemes, there will be one or more shaped queues connected to the shared queue. Thus, the input arrival curve $\alpha_{Q_i}^{\text{ATS}}(t)$ of aggregate flows before the shared queue Q_i is the sum of output arrival curves from all shaped queues q connected to Q_i,

$$\alpha_{Q_i}^{\text{ATS}}(t) = \sum_{q} \alpha_q^*(t), \tag{11}$$

where $\alpha_q^*(t)$ is from Eq. (10). Note that frames in all shaped queues q connected to the same shared queue Q_i have the same priority.

By applying $\alpha_{Q_i}^{\text{ATS}}$ and $\beta_{Q_i}^{\text{ATS}}(t)$ into Eq. (1) and Eq. (3), the upper bound of latency $D_{Q_i}^{\text{ATS}}$ and backlog $B_{Q_i}^{\text{ATS}}$ for SP flows passing through the shared queue Q_i can be given.

Service Curve $\beta_q^{\text{ATS}}(t)$ - ATS - Shaped Queue. As proved by Theorem 5 in [25], the ATS shaper is a kind of minimal interleaved regulator, which has the characteristics that placing it at the back-end of the FIFO system will not introduce extra to the worst-case delay, i.e., the worst-case delay in the combined system of the front-end FIFO system and ATS shaper is the same as the worst-case delay of the front-end FIFO system alone. Obviously, the shared queue is served in FIFO manner. Thus, a flow fed to the shaped queue q on the subsequent node will not increase the upper bound of the delay for the flow waiting in the combined element of shared queue Q_i^- on the preceding node and the shaped queue q, i.e., $d_{Q_i^-}^{\text{ATS}}(t) + d_q^{\text{ATS}}(t) \leq D_{Q_i^-}^{\text{ATS}}$, where $D_{Q_i^-}^{\text{ATS}}$ is the latency upper bound of SP flows with priority i waiting in the preceding shared queue Q_i^- and can be calculated by applying $\alpha_{Q_i^-}^{\text{ATS}}(t)$ (Eq. (11)) and $\beta_{Q_i^-}^{\text{ATS}}(t)$ (Eq. (9)) to Eq. (1). Then, for those flows transmitting from Q_i^- to q, as their lower bound of the delay in the shared queue Q_i^- is $l_q^{\min}/C$, the maximum latency D_q^{ATS} of SP flows waiting in the shaped q can be given by,

$$D_q^{\text{ATS}} = D_{Q_i^-}^{\text{ATS}} - l_q^{\min}/C. \tag{12}$$

Note that not all flows in the shared queue Q_i^- will enter into the same shaped queue q, as they may be forwarded to the other egress port of the subsequent node. Moreover, according to the ATS queuing schemes QAR1 and QAR2, flows queuing in the shaped queue q can only come from the same preceding shared queue Q_i^-.

Then, the service curve $\beta_q^{\text{ATS}}(t)$ for aggregate flows in the shaped queue q can be given by means of the burst-delay function [35]

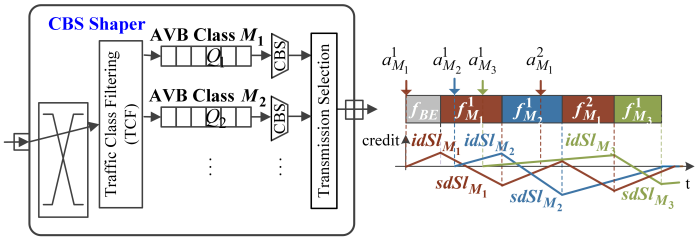
$$\beta_q^{\text{ATS}}(t) = \delta_D^q(t) \tag{13}$$

where $\delta_D^q(t)$ is the burst-delay function [41] which equals to 0 if $t \leq D$ and $+\infty$ otherwise, while $D = D_q^{\text{ATS}}$ (Eq. 12) is the delay upper bound of flows in the shaped queue q.

Input Arrival Curve $\alpha_q^{\text{ATS}}(t)$ - ATS - Shaped Queue. The input arrival curve $\alpha_q^{\text{ATS}}(t)$ of aggregate flows before reaching the corresponding shaped queue q is related to the output arrival curve $\alpha_{Q_i^-}^*(t)$ when these flows depart the preceding shared queue Q_i^-, and can be given by,

$$\alpha_{Q_i^-}^*(t) = \sum_{f \in [Q_i^-, q]} \alpha_f^{Q_i^-}(t) \oslash \delta_D^{Q_i^-}(t), \tag{14}$$

where $\alpha_f^{Q_i^-}(t) = r_f \cdot t + b_f$ is the input arrival curve of the flow f before the shared queue Q_i^-, $\delta_D^{Q_i^-}(t)$ is the burst-delay function of the delay bound $D = D_{Q_i^-}^{\text{ATS}}$ for aggregate SP flows of priority i in the preceding shared queue Q_i^-. Note that we use $f \in [Q_i^-, q]$ instead of $f \in Q_i^-$ to emphasize that not all



(a) Architecture (b) CBS forwarding frames example

Fig. 4. CBS Architecture

flows queuing in the shared queue Q_i^- will be forwarded to the same shaped queue q.

Moreover, different flows sharing the same shaped queue q cannot arrive on the shared queue Q_i at the same time, because flows sharing a common link are serialized. Thus, by taking all flows from Q_i^- to q as a group, their output arrival curve from Q_i^- can be refined by considering the constraint from the physical link with the shaping curve $\sigma^{\text{link}}(t) = C \cdot t$. According to the ATS queuing schemes QAR1 and QAR2, all flows in the shaped queue q must be from the same preceding shared queue Q_i^-. Then, the input arrival curve $\alpha_q^{\text{ATS}}(t)$ of aggregate flows before the shaped queue q is given by,

$$\alpha_q^{\text{ATS}}(t) = \alpha_{Q_i^-}^*(t) \wedge \left(\sigma^{\text{link}}(t) + l_{Q_i^-}^{\max}\right), \tag{15}$$

where $\alpha_{Q_i^-}^*(t)$ is given by Eq. (14), $x \wedge y = \min\{x, y\}$, and $l_{Q_i^-}^{\max}$ is the maximum frame size in the queue Q_i^-, which needs to be taken into account since the frame is packetized at the switch input.

By applying $\alpha_q^{\text{ATS}}(t)$ and $\beta_q^{\text{ATS}}(t)$ into Eq. 3, the upper bound of backlog B_q^{ATS} in the shaped queue q can be given.

Remark. As discussed above, the ATS shaper only plays a role in reshaping the flows, but will not increase the worst-case delay of the flow in the node transmission. Thus, the end-to-end latency bound for the flow can be obtained by summing up latency bounds $D_{Q_i}^{\text{ATS}}(t)$ only for each shared queue along its path.

3.3. Credit Based Shaper (CBS)

The Credit Based Shaper (CBS) [7] is another queuing and forwarding rule proposed for the bandwidth reservation for Audio-Video Bridging (AVB) traffic. Fig. 4(a) depicts a CBS architecture model of an output port of a node. Currently, AVB classes (i.e., Stream Reservation (SR) classes) M_i are expanded from two to more (a maximum of seven, $i \leq 7$) priorities supported by TSN [1]. Each AVB class corresponds to a FIFO queue, and has its own credit value for the CBS shaper, which is used to control the transmission of AVB frames. For each AVB Class M_i, the CBS algorithm has a credit value manipulated by two different parameters called "idleSlope" ($idSl_i$) and "sendSlope" ($sdSl_i = idSl_i - C$). For the AVB traffic of Class M_i, $idSl_i$ decides its maximum guaranteed bandwidth reservation, of which the minimum value is set according to the actual bandwidth usage of AVB Class M_i traffic.

The frame transmission based on the CBS functionality is shown using an example in Fig. 4(b). The credit is initialized to zero and is increasing with the idleSlope ($idSl_i$) when AVB frames are waiting to be transmitted due to other higher priority AVB frames or due to the negative credit and decreasing with the sendSlope ($sdSl_i$) during the transmission of an AVB frame. If the credit is positive and there are no frames waiting in the corresponding queue, then the credit is set to zero. However, if there are no frames waiting in the queue, but the credit of the corresponding queue is negative, it will increase with the idle slope until zero.

Since the standard 802.1Q [1] now supports multiple number of AVB classes, we extend the previous analysis work of supporting two classes [28] and three classes [29] to arbitrary number of AVB classes. The extension proof of credit bounds for arbitrary number of AVB classes can be found in our previous work [34]. Although [34] is the NC-based performance analysis for combined TAS and CBS, and TT transmission delays AVB traffic, AVB credits will not be affected by TT traffic if credit frozen during TT window and the protection interval ("guard band" (GB)), which is one of the cases discussed in [34].

3.3.1. Performance Analysis – CBS

Service Curve $\beta_{Q_i}^{\text{CBS}}(t)$ - CBS. The service for AVB traffic in the individual CBS architecture depends only on the credit state controlled by CBS. Different from SP traffic, AVB traffic with low priority can obtain the service even if the queues of AVB traffic of higher priority are not empty. This is because the AVB traffic cannot transmit if the CBS credit of its corresponding class is negative. The guaranteed service for multiple numbers of AVB classes M_i ($i \in [1, n_{\text{CBS}}]$),

$$\beta_{Q_i}^{\text{CBS}}(t) = idSl_i \left[t - \frac{c_i^{\max}}{idSl_i}\right]^+, \tag{16}$$

where $c_i^{\max}$ is the credit upper bound for AVB Class M_i,

$$c_i^{\max} = idSl_i \cdot \frac{\sum_{j=1}^{i-1} c_j^{\min} - l_{>i}^{\max}}{\sum_{j=1}^{i-1} idSl_j - C}, \tag{17}$$

where $l_{>i}^{\max} = \max_{j>i}\{l_{Q_j}^{\max}, l_{Q_{\text{BE}}}^{\max}\}$ is the maximum frame size in the traffic with the priority lower than priority M_i, and $c_i^{\min}$ is the lower bound of the credit of AVB Class M_i,

$$c_i^{\min} = sdSl_i \cdot \frac{l_{Q_i}^{\max}}{C}. \tag{18}$$

Input Arrival Curve $\alpha_{Q_i}^{\text{CBS}}(t)$ - CBS. The input arrival curve $\alpha_{Q_i}^{\text{CBS}}(t)$ of aggregate AVB flows of Class M_i before entering the corresponding queue Q_i of the intermediate node is related to the output arrival curve $\alpha_{Q_i^-}^*(t)$ of these flows departing the corresponding preceding queues Q_i^- connected to Q_i. The output arrival curve of aggregate flows from a preceding queue Q_i^- is,

$$\alpha_{Q_i^-}^*(t) = \sum_{f \in [Q_i^-, Q_i]} \alpha_f^{Q_i^-}(t) \oslash \delta_D^{Q_i^-}(t), \tag{19}$$

where $\alpha_f^{Q_i^-}(t)$ is the input arrival curve of the AVB flow f before Q_i^-, which needs to be iteratively calculated from the node before Q_i^- by Eq. (42) until the source node ES is reached, $\delta_D^{Q_i^-}(t)$ is the burst-delay function of the delay upper bound $D = D_{Q_i^-}^{\text{CBS}}$ for aggregate AVB flows of Class M_i in the preceding queue Q_i^-. Note that we use $f \in [Q_i^-, Q_i]$ instead of $f \in Q_i^-$ to emphasize that not all flows queuing in Q_i^- are forwarded to Q_i.

Similarly, all AVB flows from the same preceding queue Q_i^- are regarded as a group. On one hand, due to the physical link constraint $\sigma^{\text{link}}(t)$, they cannot arrive on Q_i at the same time. On the other hand, such a group of flows is also constrained by

the shaping curve $\sigma^{\text{CBS}}_{Q_i^-}(t)$ of CBS, indicating the effect of CBS on the output of AVB traffic. The CBS shaping curve $\sigma^{\text{CBS}}_{Q_i}(t)$ is constructed as the upper envelope of output accumulated bits of AVB Class M_i from Q_i in any time interval,

$$\sigma^{\text{CBS}}_{Q_i}(t) = idSl_i\left[t + \frac{c_i^{\max} - c_i^{\min}}{idSl_i}\right], \tag{20}$$

where $c_i^{\max}$ and $c_i^{\min}$ are upper and lower credit bounds respectively given by Eq. (17) and Eq. (18). Finally, the input arrival curve $\alpha^{\text{CBS}}_{Q_i}(t)$ of aggregate AVB flows of Class M_i before Q_i is given by,

$$\alpha^{\text{CBS}}_{Q_i}(t) = \sum_{Q_i^-}\left[\alpha^{*}_{Q_i^-}(t) \wedge \left(\sigma^{\text{link}}(t) + l^{\max}_{Q_i^-}\right) \wedge \left(\sigma^{\text{CBS}}_{Q_i^-}(t) + l^{\max}_{Q_i^-}\right)\right]. \tag{21}$$

The use of term $l^{\max}_{Q_i^-}$ is because the frame is packetized at the switch input.

By applying $\alpha^{\text{CBS}}_{Q_i}(t)$ and $\beta^{\text{CBS}}_{Q_i}(t)$ into Eq. (1) and Eq. (3), the upper bound of latency $D^{\text{CBS}}_{Q_i}$ and backlog $B^{\text{CBS}}_{Q_i}$ for AVB flows of Class M_i passing through the queue Q_i can be calculated.

IV. Performance Analysis of Combined Traffic Shapers

In this section, we discuss the combination of different traffic shapers. As we will show in Sect. 5.1V-A, from the perspective of latency, jitter and backlog, TAS outperforms than ATS, CBS and SP. However, TAS requires the synthesis of optimized GCLs, to which does not scale to large networks with many flows. This problem can be mitigated by combining different traffic shapers in the same switch architecture, to reduce the number of flows handled by TAS. Therefore, we believe that the coexistence between time-triggered shapers (TAS) and various event-triggered shapers (ATS, CBS, SP) will be a promising approach in the time-critical and real-time communication networks of the future, to support different performance quality requirements of applications, see also the discussion in [42]. Two combined traffic shapers investigated in the literature are non-time-triggered-CDT+ATS+CBS [35] and TAS+CBS [33], [34]. However, CDT is non-time-triggered traffic type, and the CDT model is not a standard model required by the TSN standard. Inspired by the high performance of TAS, the ATS reshaping function and the existing combined traffic shapers, we are interested in understanding the impact of ATS reshaping on the combined architectures that include TAS. Thus, in the following, we propose additional three architectures of combined traffic shaper, i.e., TAS+SP, TAS+ATS+SP, TAS+ATS+CBS. We extend the NC approach to analyze the worst-case performance of traffic under these architectures for the quantitative performance comparison in Sect. 5.2V-B.

4.1. TAS+SP / TAS+CBS

We first address the combinations without ATS. One possible combination is that of the Time-Aware Shaper (TAS) and Strict Priority (SP) queuing (i.e., TAS+SP), shown in Fig. 5(a). The combined scheduling mechanisms of TAS+SP are inherited from TTEthernet which supports Time-Triggered (TT) traffic and Rate-Constrained (RC) communication with a strict priority allocation. The difference in TSN how the TT frames are transmitted. In TTEthernet, TT communication is implemented by directly controlling the temporal behavior of each individual frame of the TT flows [43]. However, in TSN, TT communication depends on the gate control for corresponding TT queues in

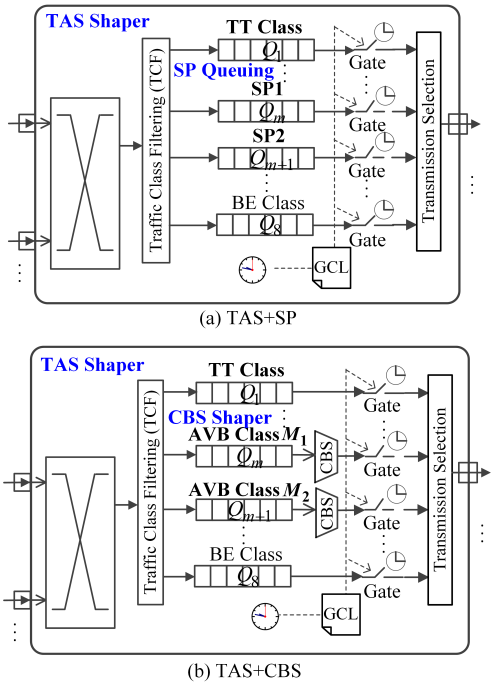


Fig. 5. TAS+SP/CBS Combined Shaper Architecture

egress ports, which requires flow or frame isolation constraints in order to achieve completely deterministic transmission [12], [13]. Another possible combination is that of TAS and the Credit Based Shaper (CBS) (TAS+CBS) [34]. The TAS+CBS architecture is presented in Fig. 5(b). In any combination, TT traffic implemented by TAS always has the highest priority. Thus, it will have the same high real-time performance as when individually used. SP/AVB traffic has the secondary priority. Different from SP scheduling, which handles the flows based on their priorities, CBS enforces a bandwidth reservation for multiple priorities of AVB traffic. CBS is used to prevent the starvation of lower-priority AVB traffic, and can tolerate a certain degree of degradation on real-time performance for high-priority traffic.

With TSN, there is a gate for each queue of egress port. Only when the gate is open, the frames in the corresponding queue can be forwarded. If more than one gate opens at the same time, the frame transmission is based on their priority. In order to keep the completely deterministic transmission for TT traffic, when an associated gate for TT traffic is open, the remaining gates for other traffic types (SP, AVB, etc.) are closed, and vice versa. Thus, lower priority traffic can be prevented from occupying the time slots reserved for TT frames. In this paper, we consider the non-preemption integration mode [4] to solve the issue when a SP/AVB frame is already in transmission at the beginning of the time slot reserved for TT traffic, as shown in Fig. 6. The non-preemption mode introduces a "guard band" (GB) interval before the TT time slot to ensure no additional delay and jitter for TT traffic. The frame is prevented from initiating transmission if there is not sufficient time for the whole frame transmission before the gate is closed. For SP/AVB traffic, the maximum GB (L^{GB}) equals to the transmission time of a maximum SP/AVB frame waiting in the corresponding queue.

For the combined traffic shapers, there is no need to re-analyze TT traffic shaped by the TAS shaper, as TT traffic is scheduled within pre-allocated time slots and is not interfered by other traffic types. However, for lower priority SP/AVB traf-

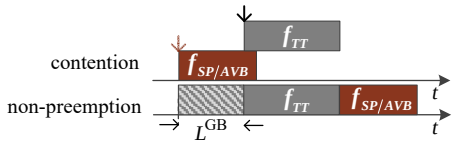


Fig. 6. Non-preemption integration modes of TT and SP/AVB Traffic

fic in the combined traffic shaper TAS+SP/TAS+CBS, the real-time performance is different from the one in the corresponding individual (SP or CBS) traffic shapers. Here, the lower priority traffic (SP/AVB) can only obtain the remaining service after TT frames are forwarded. Therefore, for the TAS+SP/CBS combined traffic shaper, it is necessary to first calculate the arrival curve related to TT traffic, i.e., the maximum accumulated bits within any time interval that could not be served for lower priority traffic (SP/AVB).

Arrival Curve $\alpha_{\mathrm{I}}^{\mathrm{TAS}}(t)$ - TAS. The construction of the arrival curve $\alpha_{\mathrm{I}}^{\mathrm{TAS}}(t)$ for TT traffic [33], [34] needs to be considered from two aspects. One is directly related to the time slots (windows), occupied by TT frames. The other is related to the integration mode, i.e., the GB interval. Whether the GB and the corresponding TT time slot are taken as a whole to construct the arrival curve $\alpha_{\mathrm{I}}^{\mathrm{TAS}}(t)$ depends on the selection of scheduling mechanisms (TAS+SP or TAS+CBS). More specifically for CBS, it depends on the credit variation (frozen or not) during GB, which will be explained in detail in Sect. 4.1.22. In the following, we use the subscript $\mathrm{I} = \{\mathrm{TT}, \mathrm{GB}+\mathrm{TT}\}$ for distinguishing two forms of the arrival curve related to TT traffic.

The time slots for TT traffic are given by GCLs for each egress port, as in the example in Fig. 2. They appear periodically according to the GCL period T_{GCL}, and the number N_{TT} of TT time slots within the GCL period is finite. It is assumed that the ith time slot occupied by TT traffic starts at time o_i^{TT} and has the time duration L_i^{TT}. Then, the relative offset between the starting times of the ith and jth time slots is $o_j^{\mathrm{TT}} - o_i^{\mathrm{TT}}$. Consequently, the arrival curve $\alpha_{\mathrm{I}}^{\mathrm{TAS}}(t)$ related to TT traffic can be given for all $t \in \mathbb{R}^+$ [34],

$$\alpha_{\mathrm{I}}^{\mathrm{TAS}}(t) = \max_{0 \le i \le N_{\mathrm{TT}}-1} \left\{ \sum_{j=i}^{i+N_{\mathrm{TT}}-1} l_{\mathrm{I},j}^{\mathrm{TAS}} \left\lceil \frac{t - o_{\mathrm{I},j,i}^{\mathrm{TAS}}}{T_{\mathrm{GCL}}} \right\rceil \right\}, \tag{22}$$

where

$$l_{\mathrm{I},j}^{\mathrm{TAS}} = \begin{cases} L_j^{\mathrm{TT}} \cdot C, & \mathrm{I} = \mathrm{TT} \\ \left(L_j^{\mathrm{TT}} + L_j^{\mathrm{GB}}\right) \cdot C, & \mathrm{I} = \mathrm{GB}+\mathrm{TT}, \end{cases}$$

and

$$o_{\mathrm{I},j,i}^{\mathrm{TAS}} = \begin{cases} o_j^{\mathrm{TT}} - o_i^{\mathrm{TT}}, & \mathrm{I} = \mathrm{TT} \\ o_j^{\mathrm{TT}} - o_i^{\mathrm{TT}} + L_i^{\mathrm{GB}} - L_j^{\mathrm{GB}}, & \mathrm{I} = \mathrm{GB}+\mathrm{TT}. \end{cases}$$

Note that L_j^{GB} is the minimum value of one maximum frame of lower priority of interest and maximum idle time slot between two consecutive TT time slots $o_j^{\mathrm{TT}} - o_{j-1}^{\mathrm{TT}} + L_{j-1}^{\mathrm{TT}}$.

4.1.1. Performance Analysis – TAS+SP

Service Curve $\beta_{Q_i}^{\mathrm{SP}}(t)$ - SP. SP traffic of different priorities competes for the leftover bandwidth after serving TT traffic. Moreover, the service SP traffic obtains also depends on the integration mode selected. Since we consider the non-preemption mode, there will be a GB before each TT window to prevent an SP frame already in transmission interfering with TT traffic. Then, in the worst-case, the time slot that SP traffic cannot occupy will be enlarged to GB + TT. SP traffic with low priority can obtain the service only when the queues of SP traffic of higher priority are empty. Then, the service curve for SP traffic with priority i ($i \in [1, n_{\mathrm{SP}}]$) in the corresponding queue Q_i can be given as follows,

$$\beta_{Q_i}^{\mathrm{SP}}(t) = C \left[t - \frac{\alpha_{\mathrm{GB+TT}}^{\mathrm{TAS}}(t)}{C} - \frac{\sum_{j=1}^{i-1} \alpha_{Q_j}^{\mathrm{SP}}(t)}{C} - \frac{l_{>i}^{\max}}{C} \right]_{\uparrow}^{+}, \tag{23}$$

where $[f(t)]_{\uparrow}^{+} = \max_{0 \le s \le t} \{f(s), 0\}$, $\alpha_{\mathrm{GB+TT}}^{\mathrm{TAS}}(t)$ is from Eq. (22) with $\mathrm{I} = \mathrm{GB}+\mathrm{TT}$, $\alpha_{Q_j}^{\mathrm{SP}}(t)$ (Eq. (24)) is the arrival curve of aggregate SP flows with the priority j higher than the priority i, and $l_{>i}^{\max}$ is the maximum frame size in traffic with the priority lower than the priority i.

Input Arrival Curve $\alpha_{Q_i}^{\mathrm{SP}}(t)$ - SP. The input arrival curve $\alpha_{Q_i}^{\mathrm{SP}}(t)$ of aggregate SP flows with the priority i before entering the corresponding queue Q_i of the intermediate node is related to the total output arrival curve $\alpha_{Q_i^-}^{*}(t)$ of these flows departing the corresponding preceding queues Q_i^- connected to Q_i and to the shaping curve $\sigma^{\mathrm{link}}(t)$ of the physical link by taking all SP flows from Q_i^- to Q_i as a group. The calculation of $\alpha_{Q_i^-}^{*}(t)$ can be done considering Eq. (19), by substituting the delay bound in $\delta_D^{Q_i^-}(t)$ with the delay upper bound $D_{Q_i^-}^{\mathrm{SP}}$ of aggregate SP flows with priority i at the preceding queue Q_i^-. Then, $\alpha_{Q_i}^{\mathrm{SP}}(t)$ can be given by,

$$\alpha_{Q_i}^{\mathrm{SP}}(t) = \sum_{Q_i^-} \left[\alpha_{Q_i^-}^{*}(t) \wedge \left(\sigma^{\mathrm{link}}(t) + l_{Q_i^-}^{\max} \right) \right]. \tag{24}$$

By applying $\alpha_{Q_i}^{\mathrm{SP}}(t)$ and $\beta_{Q_i}^{\mathrm{SP}}(t)$ in Eq. (1) and Eq. (3), the upper bound of latency $D_{Q_i}^{\mathrm{SP}}$ and backlog $B_{Q_i}^{\mathrm{SP}}$ for SP flows of priority i passing through the queue Q_i under the architecture TAS+SP can be determined.

4.1.2. Performance Analysis – TAS+CBS

Service Curve $\beta_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}(t)$ - CBS. The service for AVB traffic in the TAS+CBS architecture depends not only on the leftover service after serving TT traffic, but also on the credit state controlled by CBS. AVB traffic with different classes competes for the remaining bandwidth. When the gate for the AVB queue is open, the variation of associated credit is the same as in the case CBS is used individually, see Sect. 3.3III-C. When the gate for the AVB queue is closed, i.e., during TT transmission, the credit is frozen. Especially, during GB, the gates for all AVB queues are open without any frame transmission, however. Then, the variation of credit during GB has two cases, frozen and non-frozen, which will impact the service for AVB traffic. An example of the CBS working mechanism under the non-preemption integration mode with different assumptions on variation of credit during GB is shown in Fig. 7. The service curve for AVB Class M_i ($i \in [1, n_{\mathrm{CBS}}]$) in the corresponding queue Q_i is given by [34],

$$\beta_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}(t) = idSl_i \left[t - \frac{\alpha_{[\mathrm{M}]}^{\mathrm{TAS}}(t)}{C} - \frac{c_{i[\mathrm{M}]}^{\max}}{idSl_i} \right]_{\uparrow}^{+}, \tag{25}$$

where $\mathrm{M} = \{\mathrm{F}, \mathrm{NF}\}$ representing the choice of the credit state during GB (F — frozen credit during GB; NF — non-frozen credit during GB), $\alpha_{[\mathrm{M}]}^{\mathrm{TAS}}(t) = \{\alpha_{\mathrm{TT}}^{\mathrm{TAS}}(t), \alpha_{\mathrm{TT+GB}}^{\mathrm{TAS}}(t)\}$ from Eq. (22), and the credit upper bound $c_{i[\mathrm{M}]}^{\max} = \{c_i^{\max}, \bar{c}_i^{\max}\}$ of AVB Class M_i. Here $c_i^{\max}$ is the credit upper bound when credit is considered frozen during GB, which equals to the credit upper bound (Eq. (17)) of CBS used individually, and

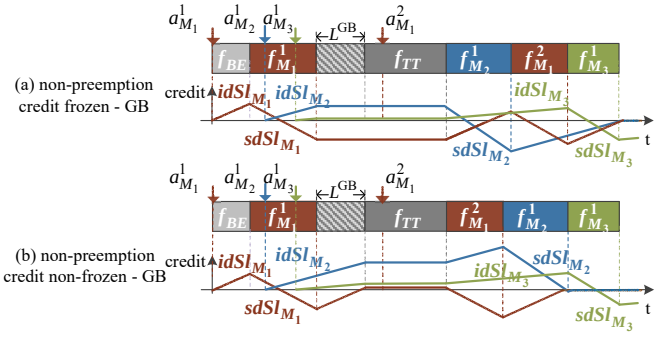


Fig. 7. CBS forwarding frames under the impact of TAS

$\overline{c}_i^{\max}$ is the credit upper bound when considering the non-frozen credit during GB [(1)],

$$\overline{c}_i(t) \le idSl_i \cdot \frac{\sum_{j=1}^{i-1} c_j^{\min} - l_{>i}^{\max} - \sigma_i^{\mathrm{GB}}}{\rho_i^{\mathrm{GB}} + \sum_{j=1}^{i-1} idSl_j - C}. \tag{26}$$

where $c_j^{\min}$ is the lower bound of credit of AVB Class M_j (Eq. (18)), and σ_i^{GB} and ρ_i^{GB} are parameters of the linear upper envelope related to GB duration and satisfy $\forall s,t \in \mathbb{R}^+, s \le t$, $C \cdot \Delta t_{\mathrm{GB}}(s,t) \le \sigma_i^{\mathrm{GB}} + \rho_i^{\mathrm{GB}} \cdot (t - s - \Delta t_{\mathrm{TT}}(s,t))$ [(2)].

The choice of expressions of $\alpha_{[\mathrm{M}]}^{\mathrm{TAS}}(t)$ and $c_{i[\mathrm{M}]}^{\max}$ depends on the credit state during GB, as follows:

- M=F: There will be a GB before each TT window to prevent an AVB frame already in transmission interfering with TT traffic. Then, the time slot that AVB traffic cannot occupy will be enlarged to GB+TT. Moreover, since credit is always frozen during GB+TT, the maximum credit value will not be affected by GB+TT time slots and equals the credit upper bound when using CBS individually. Here we take GB and TT slots as a whole and then have $\alpha_{[\mathrm{M}]}^{\mathrm{TAS}}(t) = \alpha_{\mathrm{TT+GB}}^{\mathrm{TAS}}(t)$, $c_{i[\mathrm{M}]}^{\max} = c_i^{\max}$.
- M=NF: Although there is a GB before each TT window, and no AVB traffic class can transmit during GB+TT, the credit of the corresponding AVB class will be increased during GB, however. Therefore, when deriving the service curve for AVB traffic, GB and TT slots cannot be taken as a whole as the maximum credit value will be affected by GB duration, which is not equal to the one in individually using CBS any more. Here we have $\alpha_{[\mathrm{M}]}^{\mathrm{TAS}}(t) = \alpha_{\mathrm{TT}}^{\mathrm{TAS}}(t)$, $c_{i[\mathrm{M}]}^{\max} = \overline{c}_i^{\max}$.

Input Arrival Curve $\alpha_{Q_i[\mathrm{M}]}^{\mathbf{CBS}}(t)$ - CBS. Similar to the CBS used individually, the input arrival curve $\alpha_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}(t)$ of aggregate AVB flows with priority M_i before entering the corresponding queue Q_i of the intermediate node is related to the total output arrival curve $\alpha_{Q_i^-[\mathrm{M}]}^*(t)$ of these flows in the preceding queues Q_i^- connected to Q_i, to the shaping curve $\sigma^{\mathrm{link}}(t)$ of the physical link by taking all AVB flows from Q_i^- to Q_i as a group, and to the shaping curve $\sigma_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}(t)$ (Eq. (28)) of CBS with the consideration of TAS influence,

$$\alpha_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}(t) = \sum_{Q_i^-} \left[\alpha_{Q_i^-[\mathrm{M}]}^*(t) \wedge \left(\sigma^{\mathrm{link}}(t) + l_{Q_i^-}^{\max}\right) \wedge \left(\sigma_{Q_i^-[\mathrm{M}]}^{\mathrm{CBS}}(t) + l_{Q_i^-}^{\max}\right) \right]. \tag{27}$$

where the calculation of $\alpha_{Q_i^-[\mathrm{M}]}^*(t)$ can refer to Eq. (19), and the delay in $\delta_D^{Q_i^-}(t)$ is the delay upper bound $D_{Q_i^-[\mathrm{M}]}^{\mathrm{CBS}}$ of AVB Class M_i traffic at queue Q_i^-.

The CBS shaping curve $\sigma_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}(t)$ is constructed as the upper envelope of output accumulated bits of AVB Class i from Q_i in any time interval. Its expression depends on the choice of the credit state during GB and is given by,

$$\sigma_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}(t) = idSl_i \left[t - \frac{\beta_{\mathrm{TT}}^{\mathrm{TAS}}(t)}{C} + \frac{c_{i[\mathrm{M}]}^{\max} - c_i^{\min}}{idSl_i} \right]_{\uparrow}^{+}, \tag{28}$$

where $[\mathrm{M}] = \{[\mathrm{F}], [\mathrm{NF}]\}$ represents the choice of the credit state during GB, $c_{i[\mathrm{M}]}^{\max} = \{c_i^{\max}, \overline{c}_i^{\max}\}$ with $c_i^{\max}$ from Eq. (17) and $\overline{c}_i^{\max}$ from Eq. (26), of which the expression selection depends on the credit state during GB, and $\beta_{\mathrm{TT}}^{\mathrm{TAS}}(t)$ represents the minimum amount of service obtained by TT traffic in any interval, and is given as follows,

$$\beta_{\mathrm{TT}}^{\mathrm{TAS}}(t) = \min_{0 \le i \le N_{\mathrm{TT}}-1} \left\{ \sum_{j=i}^{i+N_{\mathrm{TT}}-1} \beta_{\mathrm{TDMA}}(t+t_0, L_j^{\mathrm{TT}}) \right\}, \tag{29}$$

where

$$\beta_{\mathrm{TDMA}}(t,L) = C \cdot \max\left\{ \left\lfloor \frac{t}{T_{\mathrm{GCL}}} \right\rfloor L, t - \left\lceil \frac{t}{T_{\mathrm{GCL}}} \right\rceil (T_{\mathrm{GCL}} - L) \right\},$$

and

$$t_0 = T_{\mathrm{GCL}} - L_j^{\mathrm{TT}} - o_j^{\mathrm{TT}} + o_{i-1}^{\mathrm{TT}} + L_{i-1}^{\mathrm{TT}}.$$

By applying $\alpha_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}(t)$ and $\beta_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}(t)$ into Eq. (1) and Eq. (3), the upper bound of latency $D_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}$ and backlog $B_{Q_i[\mathrm{M}]}^{\mathrm{CBS}}$ for AVB flows of Class M_i passing through the queue Q_i under the architecture TAS+CBS can be calculated for two cases of credit during GB, respectively.

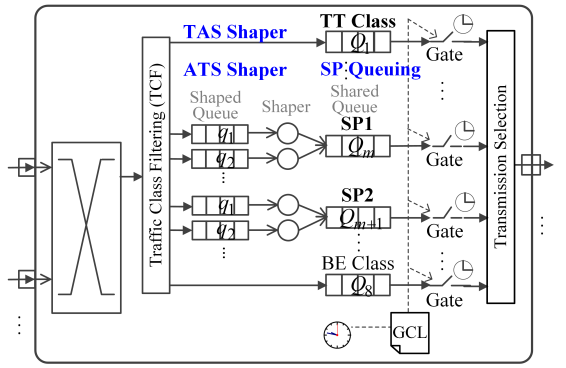
4.2. TAS+ATS+SP / TAS+ATS+CBS

An ATS shaper is a type of minimal interleaved regulator [25], used to reshape traffic before entering into the queue for each egress port of the middle node in the network. In this section, the hybrid architectures TAS+ATS+SP/CBS are presented, aiming to evaluate the reshaping influence of ATS on the real-time performance of other event-triggered shapers under the effect of the time-triggered shaper (TAS). In Sect. 5.1V-A, we will find that ATS used alone is not always superior to SP and CBS. But the advantage of the reshaping effect of ATS under the hybrid architecture is greater than that of using ATS alone as will be shown in Sect. V.

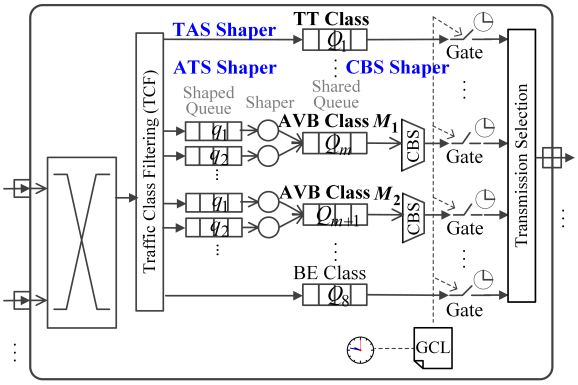
Different from the architecture of CDT+ATS+CBS proposed by [35], which assumes that CDT (control-data traffic) with the highest priority satisfies the leaky bucket model, here we consider the more general model for TAS, i.e., satisfying arbitrary time-triggered slots. For this mode, the arrival curve of TT traffic satisfies the non-linear staircase function from Eq. (22). The architectures of TAS+ATS+SP and TAS+ATS+CBS are shown in Fig. 8(a) and (b), respectively. Compared with the TAS+SP and TAS+CBS architectures in Fig. 5, there are additional shaped queues and the ATS shaping algorithm used to reshape SP/AVB flows before admitting them into their corresponding priority queues (shared queues). The queuing schemes for frames entering the shaped queues and ATS shaping algorithm are the same as the ATS used individually. Moreover, the gate operation is the same as in the architecture TAS+SP/CBS without ATS, i.e., TT traffic has the exclusive gate opening. SP/AVB frames are allowed to be transmitted

[(1)] The proof of credit upper bounds $\overline{c}_i$ for arbitrary number of AVB classes M_i can be found in [34].

[(2)] The proof of σ_i^{GB} and ρ_i^{GB} can be found from Theorem 4 and Lemma 3 in [34]



(a) TAS+ATS+SP



(a) TAS+ATS+CBS

Fig. 8. TAS+ATS+SP/CBS Combined Shaper Architecture

only when the TT gate is closed and their corresponding gates are open. As earlier, we use the non-preemption integration mode with the GB duration. Since the TT traffic controlled by the TAS has the highest priority, the traffic reshaping for SP/AVB flows by ATS will not affect the transmission of TT traffic.

In the following, we extend the NC approach to TAS+ATS+SP/CBS architectures for the quantitative performance comparison in Sect. 5.2V-B.

4.2.1. Performance Analysis – TAS+ATS+SP

Service Curve $\beta_{Q_i}^{\text{ATS+SP}}(t)$ - SP - Shared Queue. Even under the architecture of TAS+ATS+SP, the service for SP traffic with priority i ($i \in [1, n_{\text{SP}}]$) in the corresponding shared queue Q_i is only related to the TT traffic, other SP traffic with higher priority ($j < i$) and a maximum frame from lower priority traffic ($j > i$). Hence, the case is similar to the one under the TAS+SP in Sect. 4.1.11. The only difference is that, due to the ATS shaper, the rate and burst of SP traffic are reshaped and restricted before entering the shared queue. Therefore, the service curve $\beta_{Q_i}^{\text{ATS+SP}}(t)$ for SP traffic with priority i ($i \in [1, n_{\text{SP}}]$) in the corresponding shared queue Q_i under the TAS+ATS+SP architecture is similar to Eq. (23) with

$$\beta_{Q_i}^{\text{ATS+SP}}(t) = C\left[t - \frac{\alpha_{\text{GB+TT}}^{\text{TAS}}(t)}{C} - \frac{\sum_{j=1}^{i-1}\alpha_{Q_j}^{\text{ATS+SP}}(t)}{C} - \frac{l_{>i}^{\max}}{C}\right]^{+}, \tag{30}$$

but with the arrival curve $\alpha_{Q_j}^{\text{ATS+SP}}(t)$ of aggregate SP flows from Eq. (31).

Input Arrival Curve $\alpha_{Q_i}^{\text{ATS+SP}}(t)$ - SP - Shared Queue. As the output of each SP flow f departing the shaped queue q is constrained by the committed transmission rate r_f and the burst size b_f, the output arrival curve of f from the shaped queue q satisfies $r_f \cdot t + b_f$. It is also the input arrival curve of f before entering into the shared queue Q_i. Therefore, the input arrival curve $\alpha_{Q_i}^{\text{ATS+SP}}(t)$ of aggregate SP flows before the shared queue Q_i is the sum of output arrival curves from all the previous shaped queues q connected to Q_i, which is the same as the situation of ATS used individually (Eq. 11),

$$\alpha_{Q_i}^{\text{ATS+SP}}(t) = \sum_{q}\sum_{f\in q}\left(r_f \cdot t + b_f\right), \tag{31}$$

where q are all the shaped queues connected to the shared queue Q_i.

By applying $\alpha_{Q_i}^{\text{ATS+SP}}(t)$ and $\beta_{Q_i}^{\text{ATS+SP}}(t)$ into Eq. (1) and Eq. (3), we can determine the upper bound of latency $D_{Q_i}^{\text{ATS+SP}}$ and backlog $B_{Q_i}^{\text{ATS+SP}}$ for SP flows of priority i passing through the shared queue Q_i under the architecture TAS+ATS+SP.

Service Curve $\beta_q^{\text{ATS+SP}}(t)$ - SP - Shaped Queue. Since the shared queue for each SP priority in the TAS+ATS+SP architecture is served in FIFO manner, the ATS shaper will not introduce extra overheads to the worst-case delay of such a FIFO system. Thus, an SP flow fed to the shaped queue q on the subsequent node will not increase the upper bound of the delay for the flow waiting in the combined element of the shared queue Q_i^- on the preceding node and the shaped queue q, i.e., $d_{Q_i^-}^{\text{ATS+SP}}(t) + d_q^{\text{ATS+SP}}(t) \leq D_{Q_i^-}^{\text{ATS+SP}}$. Here $D_{Q_i^-}^{\text{ATS+SP}}$ is the latency bound of SP flows with priority i waiting in the preceding shared queue Q_i^- and can be calculated from the maximum horizontal deviation between $\alpha_{Q_i^-}^{\text{ATS+SP}}(t)$ (Eq. (31)) and $\beta_{Q_i^-}^{\text{ATS+SP}}(t)$ (Eq. (30)). By applying such a latency bound $D_{Q_i^-}^{\text{ATS+SP}}$ to Eq. (12), we have the maximum latency $D_q^{\text{ATS+SP}}$ of SP flows waiting in the shaped q. Then, the service curve $\beta_q^{\text{ATS+SP}}(t)$ for aggregate SP flows in the shaped queue q can be given by the burst-delay function $\beta_q^{\text{ATS+SP}}(t) = \delta_D^q(t)$, with $D = D_q^{\text{ATS+SP}}$.

Input Arrival Curve $\alpha_q^{\text{ATS+SP}}(t)$ - SP - Shaped Queue. In the TAS+ATS+SP architecture, the input arrival curve $\alpha_q^{\text{ATS+SP}}(t)$ of aggregate SP flows before entering the corresponding shaped queue q can be calculated referring to Eq. (15), by substituting the delay bound in $\delta_D^{Q_i^-}(t)$ with the delay upper bound $D = D_{Q_i^-}^{\text{ATS+SP}} = h(\alpha_{Q_i^-}^{\text{ATS+SP}}(t), \beta_{Q_i^-}^{\text{ATS+SP}}(t))$ of SP traffic with priority i in the preceding shared queue Q_i^-, where $\alpha_{Q_i^-}^{\text{ATS+SP}}(t)$ is determined by Eq. (31) and $\beta_{Q_i^-}^{\text{ATS+SP}}(t)$ is from Eq. (30).

4.2.2. Performance Analysis – TAS+ATS+CBS

Service Curve $\beta_{Q_i[\text{M}]}^{\text{ATS+CBS}}(t)$ - CBS - Shared Queue. Similarly, even under the architecture of TAS+ATS+CBS, the service for AVB traffic of Class M_i ($i \in [1, n_{\text{CBS}}]$) in the corresponding shared queue Q_i is only related to the TT traffic and credit upper bounds for the corresponding AVB traffic class. Since the credit upper bound (Eq. (17), Eq. (26)) does not depend on the arrival pattern of AVB flows, the ATS reshaping on AVB traffic does not change the ability to serve AVB traffic of different priorities in the shared queue, compared with the service capability for AVB traffic under the TAS+CBS architecture. Thus, the service curve $\beta_{Q_i[\text{M}]}^{\text{ATS+CBS}}(t)$ for AVB traffic with priority M_i ($i \in [1, n_{\text{CBS}}]$) in the corresponding shared queue Q_i under the TAS+ATS+AVB architecture is the same as the one under the TAS+AVB architecture in Sect. 4.1.22, and can be obtained by Eq. (25).

Input Arrival Curve $\alpha_{Q_i}^{\text{ATS+CBS}}(t)$ - CBS - Shared Queue. Correspondingly, by ATS reshaping, the output arrival curve

of each AVB flow f when departing the previous shaped queue q satisfies $r_f \cdot t + b_f$, which is also the input arrival curve of f before entering into the shared queue Q_i. Then, the input arrival curve $\alpha_{Q_i}^{\text{ATS+CBS}}(t)$ of aggregate AVB flows with priority i before the shared queue Q_i is the sum of output arrival curves from all the previous shaped queues q connected to Q_i, which is the same as the case of ATS used individually (Eq. 11).

By applying $\alpha_{Q_i}^{\text{ATS+CBS}}(t)$ and $\beta_{Q_i[\text{M}]}^{\text{ATS+CBS}}(t)$ into Eq. (1) and Eq. (3), we can determine the upper bound of latency $D_{Q_i[\text{M}]}^{\text{ATS+CBS}}$ and backlog $B_{Q_i[\text{M}]}^{\text{ATS+CBS}}$ for SP flows of priority M_i passing through the shared queue Q_i under TAS+ATS+CBS.

Service Curve $\beta_{q[\text{M}]}^{\text{ATS+CBS}}(t)$ - CBS - Shaped Queue. Similarly, an AVB flow fed to the shaped queue q on the subsequent node will not increase the upper bound of the delay for the flow waiting in the combined element of shared queue Q_i^- on the preceding node and the shaped queue q, i.e., $d_{Q_i^-}^{\text{ATS+CBS}}(t) + d_q^{\text{ATS+CBS}}(t) \leq D_{Q_i^-[\text{M}]}^{\text{ATS+CBS}}$, due to fact that each AVB class in the shared queue under the TAS+ATS+CBS architecture also obeys the FIFO order. Here $D_{Q_i^-[\text{M}]}^{\text{ATS+CBS}}$ is the latency bound of AVB flows of Class M_i in the preceding shared queue Q_i^- and can be calculated from the maximum horizontal deviation between $\alpha_{Q_i^-}^{\text{ATS+CBS}}(t)$ and $\beta_{Q_i^-[\text{M}]}^{\text{ATS+CBS}}(t)$. By applying such a latency bound $D_{Q_i^-[\text{M}]}^{\text{ATS+CBS}}$ to Eq. (12), we have the maximum latency $D_{q[\text{M}]}^{\text{ATS+CBS}}$ of AVB flows waiting in the shaped q. Then, the service curve $\beta_{q[\text{M}]}^{\text{ATS+CBS}}(t)$ for aggregate AVB flows in the shaped queue q can be given by the burst-delay function $\beta_{q[\text{M}]}^{\text{ATS+CBS}}(t)$, with $D = D_{q[\text{M}]}^{\text{ATS+CBS}}$.

Input Arrival Curve $\alpha_{q[\text{M}]}^{\text{ATS+CBS}}(t)$ - CBS - Shaped Queue. In the TAS+ATS+CBS architecture, in addition to the total output arrival curve $\alpha_{Q_i^-}^*(t)$ of the aggregate AVB flows departing the shared queue Q_i^- to the shaped queue q and the shaping curve $\sigma^{\text{link}}(t)$ of the physical link by taking all AVB flows from Q_i^- to q as a group, the input arrival curve $\alpha_{q[\text{M}]}^{\text{ATS+CBS}}(t)$ of aggregate AVB flows before entering the shaped queue q is also related to the CBS shaping curve $\sigma_{Q_i^-[\text{M}]}^{\text{ATS+CBS}}(t)$ in Eq. (28). We have $\alpha_{Q_i^-}^*(t)$ from Eq. (14), by substituting the delay bound in $\delta_D^{Q_i^-}(t)$ with the delay upper bound $D = D_{Q_i^-[\text{M}]}^{\text{ATS+CBS}} = h(\alpha_{Q_i^-}^{\text{ATS+CBS}}(t), \beta_{Q_i^-[\text{M}]}^{\text{ATS+CBS}}(t))$ of AVB traffic in the preceding shared queue Q_i^-. Then, $\alpha_{q[\text{M}]}^{\text{ATS+CBS}}(t)$ is given by,

$$\alpha_{q[\text{M}]}^{\text{ATS+CBS}}(t) = \alpha_{Q_i^-}^*(t) \wedge \left(\sigma^{\text{link}}(t) + l_{Q_i^-}^{\max}\right) \wedge \left(\sigma_{Q_i^-[\text{M}]}^{\text{ATS+CBS}}(t) + l_{Q_i^-}^{\max}\right). \tag{32}$$

Note that due to the ATS queuing schemes QAR1 and QAR2, flows queuing in the shaped queue q can only come from the same preceding shared queue Q_i^-.

V. Performance Comparison Evaluation

In this section, in order to compare the performance evaluation of individual traffic shapers and their combinations, we use a large set of synthetic test cases with different topologies and a realistic test case, i.e., the Orion Crew Exploration Vehicle (CEV) from NASA [44].

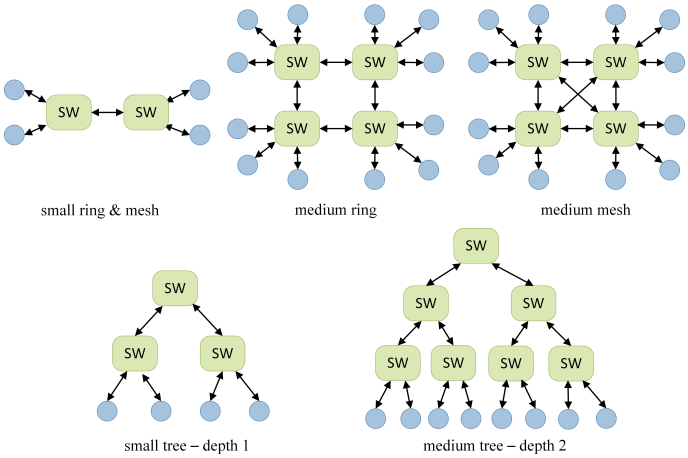


Fig. 9. Network topologies of synthetic test cases

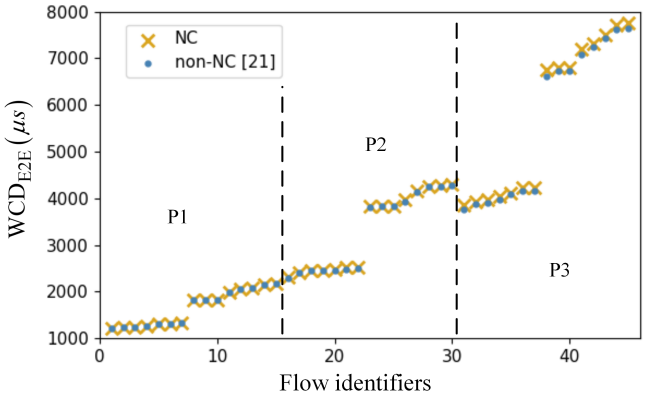


Fig. 10. Comparison of NC and non-NC methods for ATS evaluation

5.1. Individual Traffic Shapers

5.1.1. Comparison of NC and non-NC approaches for ATS evaluation

Before discussing the various traffic shapers, we first compare the two different methods used for ATS evaluation, i.e. the Network Calculus (NC) used in this article and a non-NC approach proposed in [21]. By comparing the upper bound of the delay obtained by the two methods, the latency bound for a flow f in an egress port calculated by NC is ΔD_f^Q more pessimistic than the result calculated by the non-NC approach [21],

$$\Delta D_f^Q = \max_{\forall f' \in Q(f)} \left(\frac{l_{f'}}{C - \sum_{f'' \in Q_H} r_{f''}} - \frac{l_{f'}}{C} \right), \tag{33}$$

where l'_f is the frame size of a flow with the same priority level of the flow f of interest, l''_f is the frame size of flows with higher priority than f, and $r_{f''}$ is the committed burst size of the flow f'' supported by ATS.

For the evaluation of two approaches, we use a synthetic test case where the topology is a medium mesh (Fig. 9), including 45 flows with 3 priorities. The average traffic load is around 70%, and physical link rate is set to 100 Mb/s. We show a comparison of the two methods in Fig. 10, where the value on the x-axis represents identifiers of each flow and the y-axis shows the upper bound of end-to-end latency in microseconds. The obtained results are grouped by priority, denoted vertical dotted lines, and sorted in increasing order by results within each priority. As we can see from Fig. 10, the performance evaluation by the NC analysis of ATS is very close to the

TABLE II
STATISTICAL HOPS AND TRAFFIC LOAD FOR 100 TEST CASES

	SRM	MR	MM	ST	MT
Average Hops	2.7	4.2	3.8	3.5	5.5
Average Traffic Load	28.9%	20.5%	17.4%	29.0%	19.7%
Max Traffic Load	47%	40%	38%	47%	30%
Min Traffic Load	13%	8%	6%	13%	10%

analysis from [21], which is the as expected according to Eq. (33). We note that with a decrease in the priority, the gap between the two will increase slightly. This is because the denominator of the first term of Eq. (33) is related to the sum of the rates of all high-priority flows. The lower the priority of the flow of interest, the greater the rate accumulated by the high-priority flow. Thus, the first term of Eq. (33) is becoming larger. For example, for the highest priority flows, the results from two approaches are completely the same. For the lower priority 1 and 2, the evaluation results calculated by NC are 0.6% and 1.4% slightly more pessimistic on average than the results by non-NC approach, respectively.

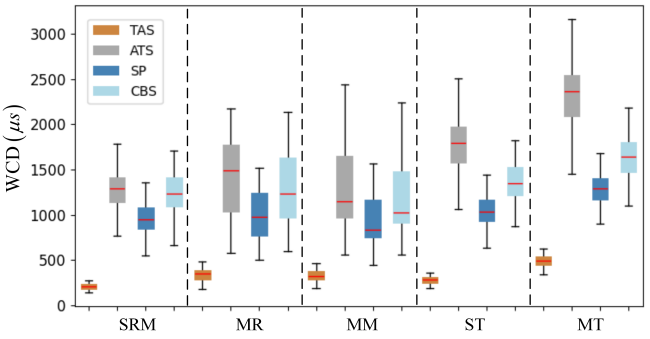
However, the non-NC approach proposed by [21] is focused on ATS in isolation and thus it is not applicable and cannot be extended to combinations of traffic shapers. Hence, in this paper, we consider the Network Calculus approach for evaluating various traffic shapers and their combinations. In the following, all the evaluation results are based on the Network Calculus approach introduced in this article.

5.1.2. Performance comparison among TAS, ATS, SP and CBS

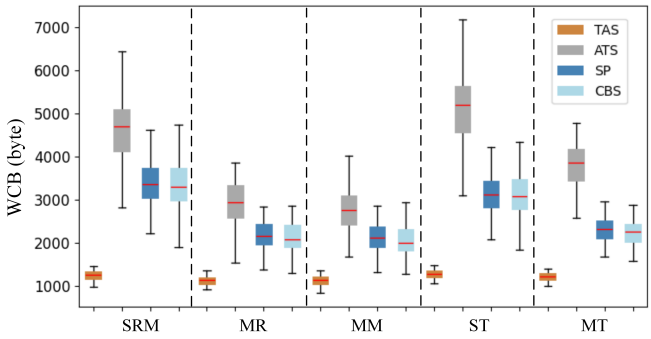
In the first set of experiments, we are interested to compare the performance from the perspective of the upper bounds of end-to-end latency, jitter and backlog without the frame loss for each individual traffic shapers (including TAS, ATS, CBS and SP) under different network topologies. The network topologies are respectively small ring & mesh (SRM), medium ring (MR), medium mesh (MM), small tree-depth 1 (ST) and medium tree-depth 2 (MT) which are inspired from industrial application requirements [43], as shown in Fig. 9. There are 100 test cases (TCs) randomly generated. For each test case, there are 15 flows. The frame size l_f of each flow is randomly chosen between the minimum (64 bytes) and the maximum (1,522 bytes) Ethernet frame size, and flows can be periodic or sporadic[(3)]. For the periodic flow, the periods are uniformly selected from the set $T_f = \{1,000, 2,000, 5,000, 1,0000\}$ μs. For the sporadic flows, it is assumed that each flow satisfies the leaky bucket model with the burst $b_f = l_f$ and rate $r_f = l_f/T_f$. Since TT flows manipulated by the TAS have no priority division, it is assumed that all flows are assigned to the same priority level for each use case in this experiment. The GCLs for TAS is generated according to [13]. All the test cases are applied on the above five topologies, respectively. The routes of flows are generated according to the routing optimization strategy proposed for TT traffic [13]. For each test case, we considered average hops of flows and average traffic load under each topology. Table II gives the statistics over the 100 test cases under each topology. The physical link rate is set to C=100 Mb/s.

For each test case under a given topology, we evaluate the quality of service for different individual traffic shapers, i.e., TAS, ATS, CBS and SP, respectively. By applying the NC approach, we can get the two evaluation metrics of each flow under different individual traffic shapers, i.e., the upper bound of end-to-end delay and jitter; and one evaluation metric for each egress port, i.e., the upper bound of the backlog without the frame loss. Fig. 11 presents the performance evaluation of the network from the perspective of the upper bounds of end-to-end delay, backlog and jitter of different individual traffic shapers under different topologies. For each test case, we use the average value of the corresponding evaluation metric of all flows to represent the metric value under the current test case. Therefore, for each individual traffic shaper under each topology, we obtain 100 values of the corresponding metric, and we use box plots to present these results. Fig. 11(a) shows the evaluation of the end-to-end latency bounds. The x-axis represents different topologies, and the y-axis shows the upper bound of the end-to-end latency (WCD) in microseconds. Fig. 11(b) and Fig. 11(c) show the evaluation results on upper bounds of worst-case backlog (WCB) in bytes and jitter (WCJ) in microseconds, respectively.

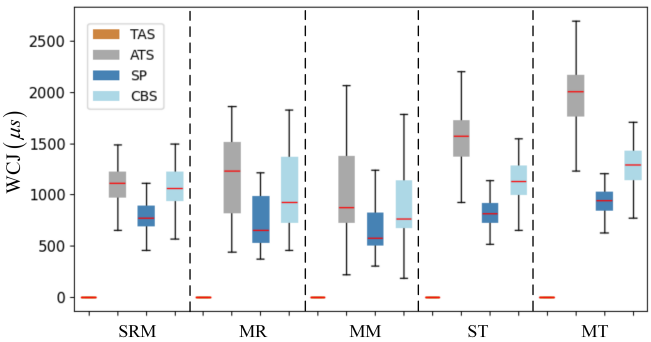
As can be seen in the figure, TAS performs best with the lowest latency, backlog and provides zero-jitter. Such performance



(a) End-to-end latency bounds



(b) Backlog bounds



(c) Jitter bounds

Fig. 11. Comparison of different individual traffic shapers under different topologies

[(3)]Flows served by the TAS are periodic, and ATS and AVB support both periodic and sporadic flows.

TABLE III
DIFFERENCE RATIO ON METRICS OF TWO INDIVIDUAL TRAFFIC SHAPERS

		(CBS − SP) /SP	(ATS − SP) /SP	(ATS − CBS) /CBS
Average WCD	SRM	30.7%	34.1%	2.8%
	MR	28.2%	43.9%	12.5%
	MM	26.2%	35.7%	7.7%
	ST	31.2%	72.3%	31.4%
	MT	27.3%	79.1%	40.8%
Average WCB	SRM	-1.3%	38.0%	39.9%
	MR	-2.5%	34.7%	38.1%
	MM	-3.2%	31.0%	35.4%
	ST	-0.8%	65.0%	66.5%
	MT	-3.3%	64.4%	70.0%
Average WCJ	SRM	37.1%	41.4%	3.4%
	MR	38.0%	60.7%	16.6%
	MM	30.3%	43.0%	10.4%
	ST	39.5%	91.9%	37.7%
	MT	37.3%	108.3%	51.9%

is in line with expectations. Since TAS realizes a completely deterministic time-triggered transmission through flow-based scheduling, it avoids the collision of frames of its own traffic type and avoids the collision with frames of other traffic types as well. Thus, flows shaped by TAS can achieve ultra-low latency, backlog and jitter. For the other three traffic shapers, i.e., ATS, SP and CBS, the overall trend of their performance comparison can be inferred from the figure. However, it is difficult to see their comparison on an individual test case from the figure. Thus, we additionally use the difference ratio

$$X_i = (X_i^{Y_1} - X_i^{Y_2})/X_i^{Y_2} \tag{34}$$

to capture the performance comparison of traffic shapers Y_1 and Y_2 ($Y \in \{\text{ATS}, \text{CBS}, \text{SP}\}$), where X_i can represent the end-to-end latency bound or jitter bound of the flow f_i or the upper bound of the backlog for the queue Q_i at the egress port. Then, we take the average value $X = average(X_i)$ as the comparison result of two traffic shapers under the current use case. The comparison results are shown in Table III.

Regarding the delay and jitter upper bound, SP performs better than that CBS, as shown in Fig. 11(a) and (c), and the "Average WCD" and "Average WCJ" of the third column in Table III. This is because CBS has a bandwidth reservation mechanism for traffic. In our case, only 75% of the bandwidth is available for AVB traffic. Therefore, compared to SP, the service bandwidth obtained for flows shaped by CBS is lower. Concerning the backlog bounds, CBS may perform better than SP, as presented in Fig. 11(b) and the "Average WCB" of the third column in Table III. This is because although the waiting time of the flow in the corresponding priority queue has been prolonged by CBS through controlling the credit, it however reduces the long-term rate of arrival of flows. Thus, it is possible that the backlog upper bounds of AVB traffic are lower than for SP traffic.

What is more interesting is that, for all the current use cases, the performance of ATS is not better than SP and CBS, a finding that has not been reported in the related work. Due to the reshaping function of ATS which avoids burstiness cascades, we initially hypothesized that the ATS would improve the performance of flows. However, as it can be observed in Fig. 11(a), (b), (c) and the fourth and fifth columns in Table III, in terms of upper bounds of delay, backlog or jitter, ATS did not perform superiorly.

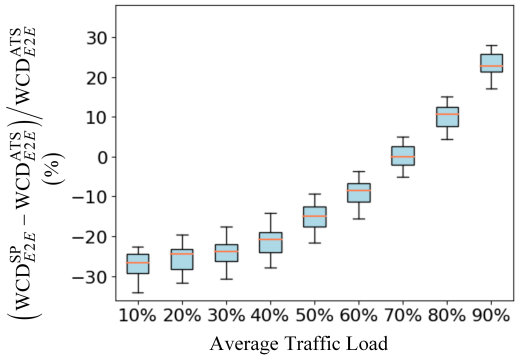
In the following, we will only focus on the comparison of ATS and SP, as ATS compared with CBS can be inferred from the comparison of SP and CBS discussed above. According to the current results, we can draw a conclusion and a hypothesis. The conclusion is that the advantages of ATS are getting worse with the increase of the concentration of flows transmission and the number of hops (under the same concentration of flows transmission). This is because that the more concentrated are the flows transmitted in the network, the more obvious is the serialization of flows in transmission, which increases the determinism of traffic transmission at subsequent nodes. Moreover, under the case of the constant concentration of flows transmission, as the number of hops increases, serialization leads to the same determinism of traffic transmission, but the time cost on ATS reshaping increases accordingly. The hypothesis is that as the load increases, the advantages of ATS will increase, which will be discussed in the next section. The reasoning behind this hypothesis is that the time cost of ATS reshaping traffic cannot offset the queuing time of traffic with the burst cascade without ATS. Thus, ATS has a negative effect when the traffic load has not reached a certain level. Therefore, it is reasonable to infer that when the traffic load increases, the impact of ATS will gradually turn to positive.

Next, we analyze the results on column 4 in Table III. Since each test case is applied under five topologies, for some use case, the five topologies have the same flows, but just with different routes. Moreover, the traffic load in Table II reflects the degree of dispersion of flows transmission in the network. For example, the traffic load under the MM topology is the lowest. This is because the MM has the largest number of selectable paths, and thus the flows are more dispersed than the traffic in other topologies. Hence, although the traffic load in MM is lower than that in SRM, the performance comparison between ATS and SP is not much different from that in SRM. Compared with MR, the traffic load in MM is close to that of MR, but the transmission of traffic in MR is more concentrated than that in MM. Therefore, the advantage of ATS under the MM topology is higher than under the MR topology. For the ST topology, although its traffic load is close to the traffic load in SRM, its number of hops is higher, which leads to more times of ATS reshaping along the flow's route. Thus, compared with SRM, the performance of ATS under the ST topology is far worse than SP. A similar explanation can be extended to the results for the MT topology.

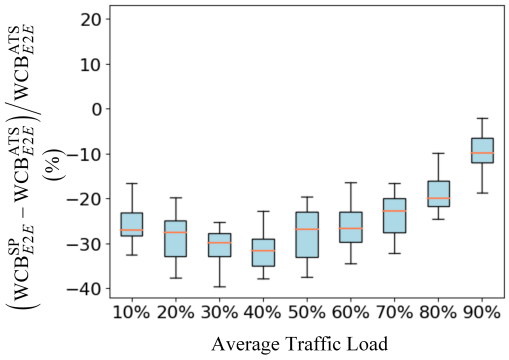
5.1.3. Comparison between ATS and SP under changing traffic load

In the previous section, the traffic load of all test cases is in the range from 6% to 47%, and in all these test cases, ATS does not show its advantages in real-time performance compared with TAS, CBS and SP. In order to rule out the influence of traffic dispersion degree and number of hops under different topologies, for the second set of experiments, we chose the same topology (MM) to study the comparative performance between ATS and SP mechanisms when the average traffic load is increasing from 10% to 90%. We used 20 random synthetic test cases under each traffic load. The frame sizes and intervals are selected in the same way as for the use cases in the previous section. As the comparison trend of individual shapers on the upper bounds of latency and jitter is very similar, we will only show the comparison of the upper bounds of delay and backlog in the following.

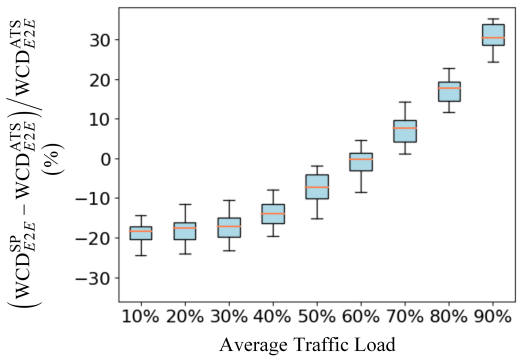
We still use the difference ratio from Eq. 34 to represent the performance comparison result of ATS and SP under the current use case, where X_i can represent the upper bound of



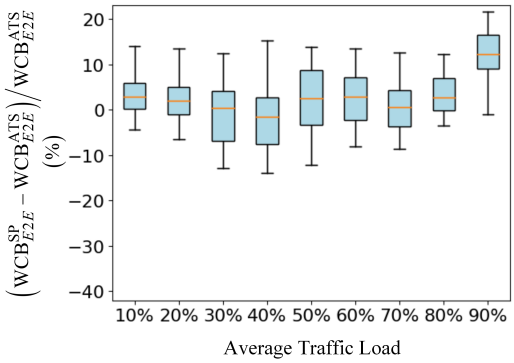
(a) End-to-end latency bounds (no interference)



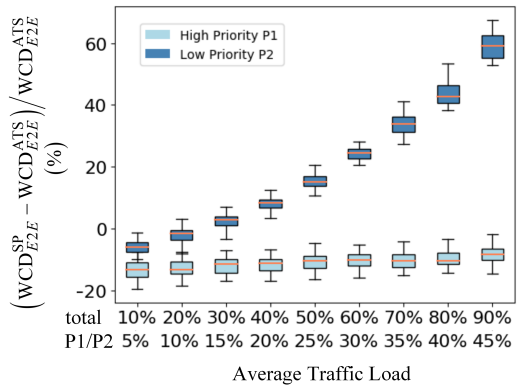
(b) Backlog bounds (no interference)



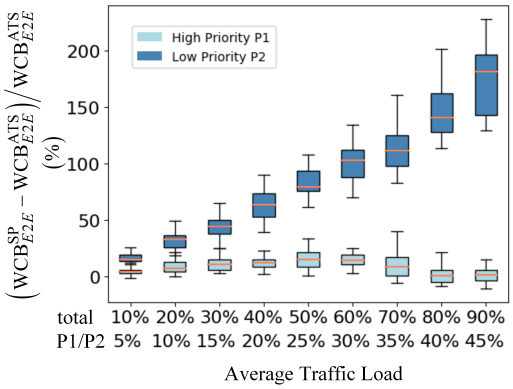
(c) End-to-end latency bounds ($l_{\mathrm{BE}} = 1522$ B)



(d) Backlog bounds ($l_{\mathrm{BE}} = 1522$ B)



(e) End-to-end latency bounds (two priorities)

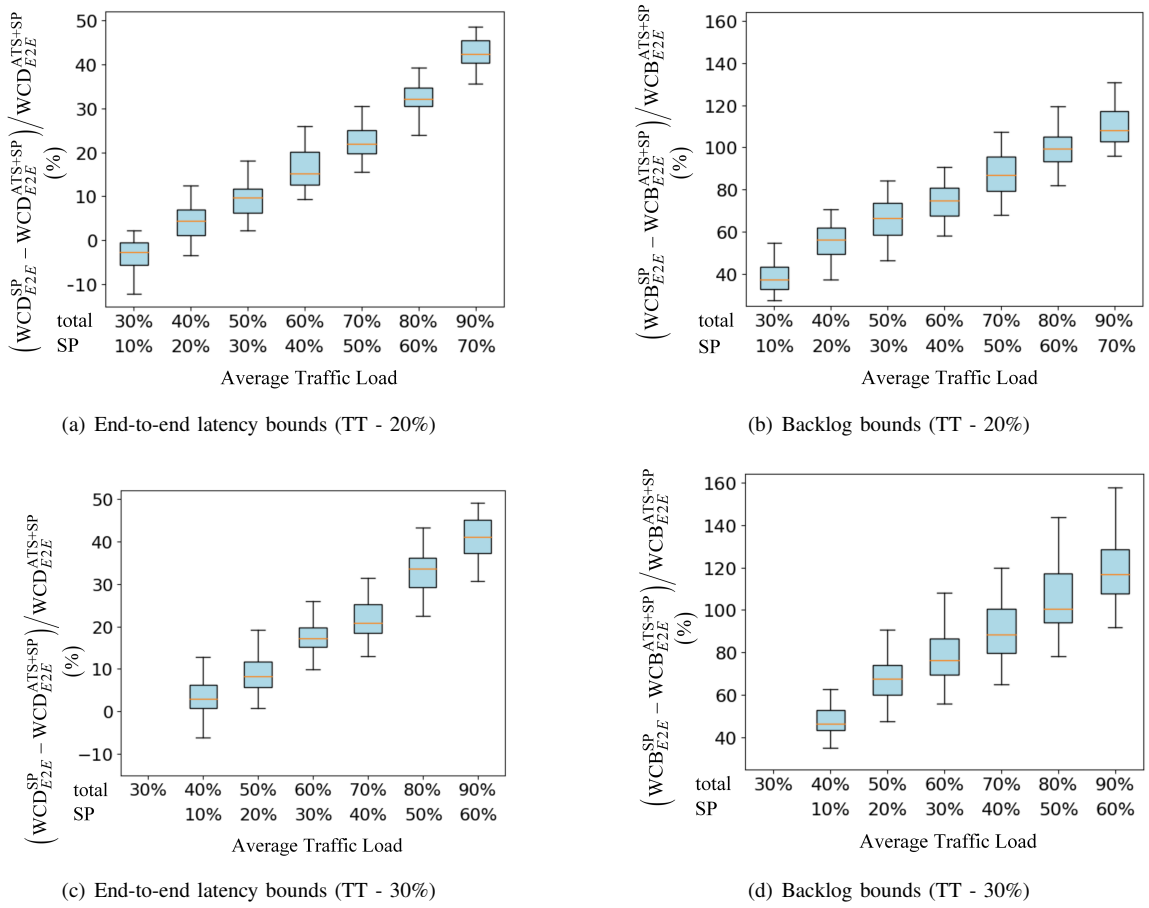


(f) Backlog bounds (two priorities)

Fig. 12. Comparison of ATS and SP under different traffic load

the end-to-end latency of flow f_i or the upper bound of the backlog for the queue Q_i at egress port.

In order to test the performance of ATS in isolation, we first assume that all flows have the same priority, and there is no interference of other traffic type. This situation can be also similar to the network adopting the preemption integration mode. The comparison results of ATS versus SP are shown in Fig. 12(a) and (b). As it can be seen, for the upper bound of end-to-end delay (Fig. 12(a)), with the increasing of average traffic load, the performance relationship between ATS and SP changes. When the average traffic load is lower than 70%, SP performs better than ATS. ATS shows its superiority only when the average traffic load increases to more than 70%. For the upper bound of the backlog (Fig. 12(b)), the difference ratio between ATS and SP does not change significantly, and when considering ATS used individually the backlog performance is always inferior to that of SP. It is furthermore assumed that all flows still have the same priority, but there exists the interference of BE traffic with the maximum Ethernet frame size of $1,522$ Bytes. If the preemption integration mode is considered, the compared results will be similar to the discussion above (Fig. 12(a), (b)). If the non-preemption integration mode is taken into account (as considered in this paper), there will be at most one BE frame interference when the flow of interest obtains the service. The results with and without ATS reshaping are shown in Fig. 12(c) and (d). As can be observed from the figure, low-priority non-preemptible frames have no significant impact on the performance with and without ATS from the perspective of upper bounds of end-to-end delays. Latency performance comparison results still mainly depend on the traffic load. However, the performance with ATS and without ATS from the perspective of backlog upper bounds is significantly changed. This is because the non-preemptible frames from low priority traffic have larger impact on the backlog performance of egress ports without ATS reshaping compared to ports with ATS reshaping. From here we can also



(a) End-to-end latency bounds (TT - 20%)

(b) Backlog bounds (TT - 20%)

(c) End-to-end latency bounds (TT - 30%)

(d) Backlog bounds (TT - 30%)

Fig. 13. Comparison of TAS+ATS+SP and TAS+SP under different traffic load

see that the backlog bounds comparison between individual ATS and SP (one priority) is more related to the frame length.

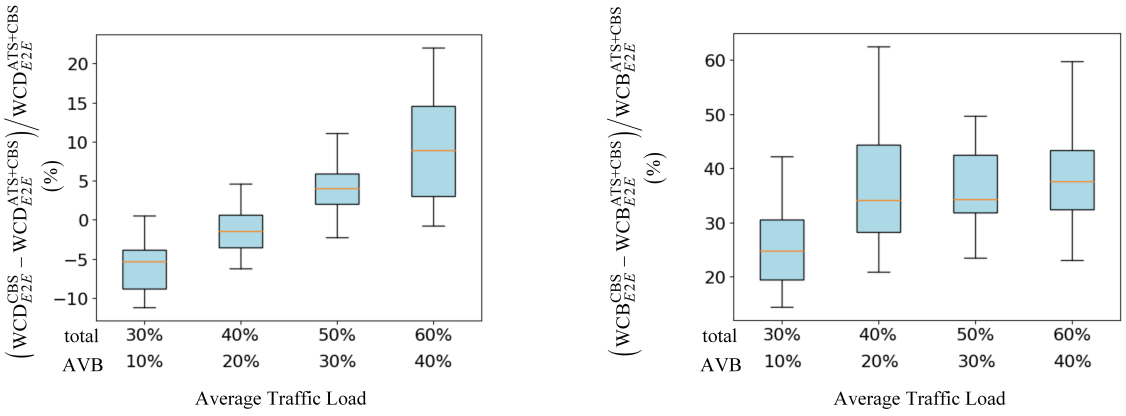
Moreover, we assign high and low priorities to the traffic, and the high and low priority traffic each account for 50% of the overall traffic load. The comparison results of ATS vs SP in terms of upper bound of the delays and backlogs for the high and low priority traffic, with and without reshaping by ATS, are shown in Fig. 12 (e) and (f), respectively, using light blue and dark blue box plots. It can be seen from the figure that for the high priority traffic, the results are similar to the top 40% compared results in Fig. 12 (c) and (d), and similarly ATS does not show its superiority for high-priority traffic. This is because the transmission of the high priority traffic will only be interfered by at most one low priority frame whose frame length ranges from the minimum (64 bytes) to the maximum ($1,522$ bytes) Ethernet frame size, which is same as the case if all flows have the same priority and are interfered by a BE frame. Moreover, the maximum average load of the high-priority traffic is only 45%. Nevertheless, for low priority, ATS shows its performance advantage from the perspective of latency bounds when the overall average traffic load reaches 30% while low priority average traffic load reaches 15%. From the perspective of backlog, ATS reshaping for low priority traffic always shows its superiority when the overall average traffic load is increasing from 10% to 90% (while the average traffic load for low priority is from 5% to 45% accordingly). The reason is that the low priority traffic can only be transmitted when the high priority queue is empty so that the burst cascade of low priority flows is more obvious than that of high priority flows. It also means that the time cost of reshaping low priority traffic by ATS is lower than the waiting time caused by the burst cascade of low-priority traffic.

Therefore, when the ATS is used individually, its positive effect on latency upper bounds can be highlighted only when the average traffic load in the network reaches a certain high level around 70% to 80%, as in Fig. 12(a) and (c). Moreover, whether it has a positive impact on the upper limit of the backlog is related to the size of the interference frame of other traffic types. The larger size of the interfering frame, the more positive impact is shown by ATS, as in Fig. 12(b) and (d). When ATS is used for traffic of multiple priorities, ATS shows a positive impact on both latency and backlog upper bounds for low priority traffic even if the average low priority traffic load is not high. However, its impact on the high priority traffic is still negative, see Fig. 12(e) and (f). Based on the above findings, this is the reason why we believe that the combined use of ATS with TAS will make ATS play a more active role. At the same time, TAS will perfectly maintain its advantages of ultra low latency and jitter. In the next section, we will see that ATS has a positive role when combined with other traffic shapers.

5.2. Combined Traffic Shapers

5.2.1. Evaluation on Synthetic Test Cases

In this section, we are interested in the influence of ATS on the real-time performance of combined traffic shapers. First, the architectures of TAS+ATS+SP and TAS+SP without ATS are compared. In the first set of experiment, it is assumed that there is 20% of TT traffic achieving deterministic transmission based on the TAS, and 10% to 70% of traffic is SP. Therefore, the overall average traffic load in the network is 30%-90%. For each traffic load, we have used 20 randomly generated test



(a) End-to-end latency bounds

(b) Backlog bounds

Fig. 14. Comparison of TAS+ATS+CBS and TAS+CBS under different traffic load

cases. In order to fairly compare with ATS used individually, we still assume that all flows reshaped by ATS have the same priority and with no interference of other traffic type of lower priority (BE for example).

As can be seen from Fig. 13(a) and (b), similar to ATS used individually, with the increase of average traffic load, the performance of traffic shaped by ATS is gradually improved. However, with the influence of TT traffic implemented by TAS, it is obvious that the optimization effect of ATS reshaping is better than that of individual ATS traffic shaper, including both upper bounds of delay and backlog. For the end-to-end latency bounds, ATS traffic shaper used individually performs better only when the average traffic load is above 70%. However, in the combined traffic shaper TAS+ATS+SP, as long as the average traffic load shaped by ATS reaches 20% (overall average load is 40%), the ATS traffic shaper is superior. For the backlog bounds, the ATS traffic shaper used individually has not shown its advantage, even at a 90% load. However, for the combined traffic shaper TAS+ATS+SP, ATS shows its superiority as long as the average traffic load shaped by ATS is above 10% (overall average load is 30%). The performance advantage of ATS in combination with TAS is similar to the performance advantage of reshaping for low priority traffic when ATS is used individually. This is because TT traffic based on TAS has fixed transmission time slots and has the highest priority, and other traffic types cannot use the time slots allocated to TT traffic. The existence of TT traffic will greatly increase the possibility of mutual interference and backlog of other types of traffic. Then, the burst cascade of the flow on its route will be increased. Therefore, the time used by the ATS reshaping is lower than the waiting time caused by the bursty traffic. Furthermore, we increase the average load of TT traffic to 30%, while there is 10% - 60% of SP traffic, thus the overall average traffic load in the network is 40% to 90%. Similarly, for each traffic load, there are 20 test cases generated randomly. The compared results of TAS+ATS+SP and TAS+SP are shown in Fig. 13(c) and (d), respectively. By comparing with 20% of TT traffic load, it is found that with the increasing TT traffic load, the positive impact of ATS is not increasing significantly.

Next, we will compare the performance of TAS+ATS+CBS and TAS+CBS without ATS. TT traffic load is still 20%. In addition, we consider the average load of AVB traffic is from 10% to 40%, as only 75% of bandwidth is reserved for AVB traffic, which also includes the bandwidth occupied

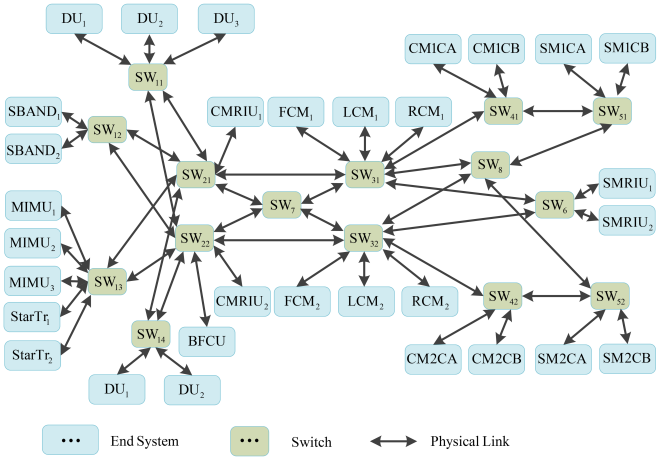
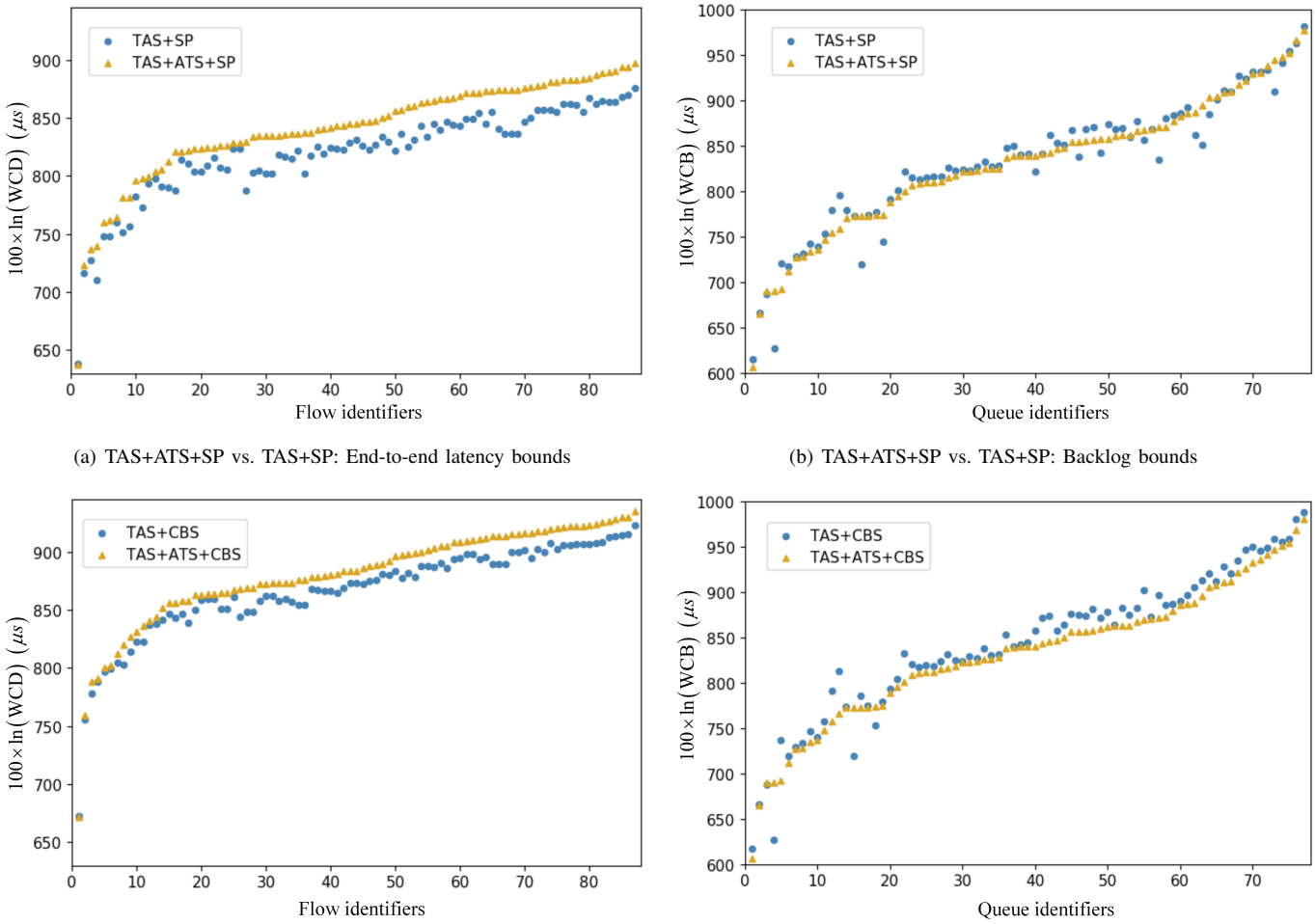


Fig. 15. Network topology of Orion CEV

by TT traffic. In addition, it is also necessary to ensure that the maximum link load in the network does not exceed 75%. Similarly, we have used 20 randomly generated test cases for each traffic load. The comparison results are shown in Fig. 14. Concerning the upper bounds of end-to-end latency and backlog, ATS also shows its superiority as long as the average load of AVB traffic is above 20% and 10%, respectively, under the TAS+ATS+CBS combination, which is similar to the case with TAS+ATS+SP, as shown in Fig. 13 (a) and (c). But the difference is that with the AVB traffic load increasing, although the optimization of the delay performance with the ATS in the architecture TAS+ATS+CBS is also increasing, it is not as obvious as that in TAS+ATS+SP. For the upper bounds of backlog, the optimization effect of ATS basically does not increase with the traffic load increasing. This is because CBS is a bandwidth reservation service. CBS itself implements a fairer service by controlling credit, thereby reducing the long-term rate of arrival of flows. Therefore, the optimization effect of ATS itself in the TAS+ATS+CBS architecture is weakened.

5.2.2. Evaluation on the Realistic Test Case

In the last experiment, we use the realistic case study the Orion Crew Exploration Vehicle (CEV) from NASA [44]. The test case topology is shown in Fig. 15. The Orion CEV case has 31 ESes, 15 SWs, 188 dataflow routes connected by physical link transmitting at 100 Mbps. In the last set of experiments, we are interested to see the effect of ATS on



(a) TAS+ATS+SP vs. TAS+SP: End-to-end latency bounds

(b) TAS+ATS+SP vs. TAS+SP: Backlog bounds

(c) TAS+ATS+CBS vs. TAS+CBS: End-to-end latency bounds

(d) TAS+ATS+CBS vs. TAS+CBS: Backlog bounds

Fig. 16. Comparison of combined traffic shapers under the Orion CEV $\text{TC}_{\text{Orion1}}$ (TT - 1.5%, SP/AVB with 1 priority)

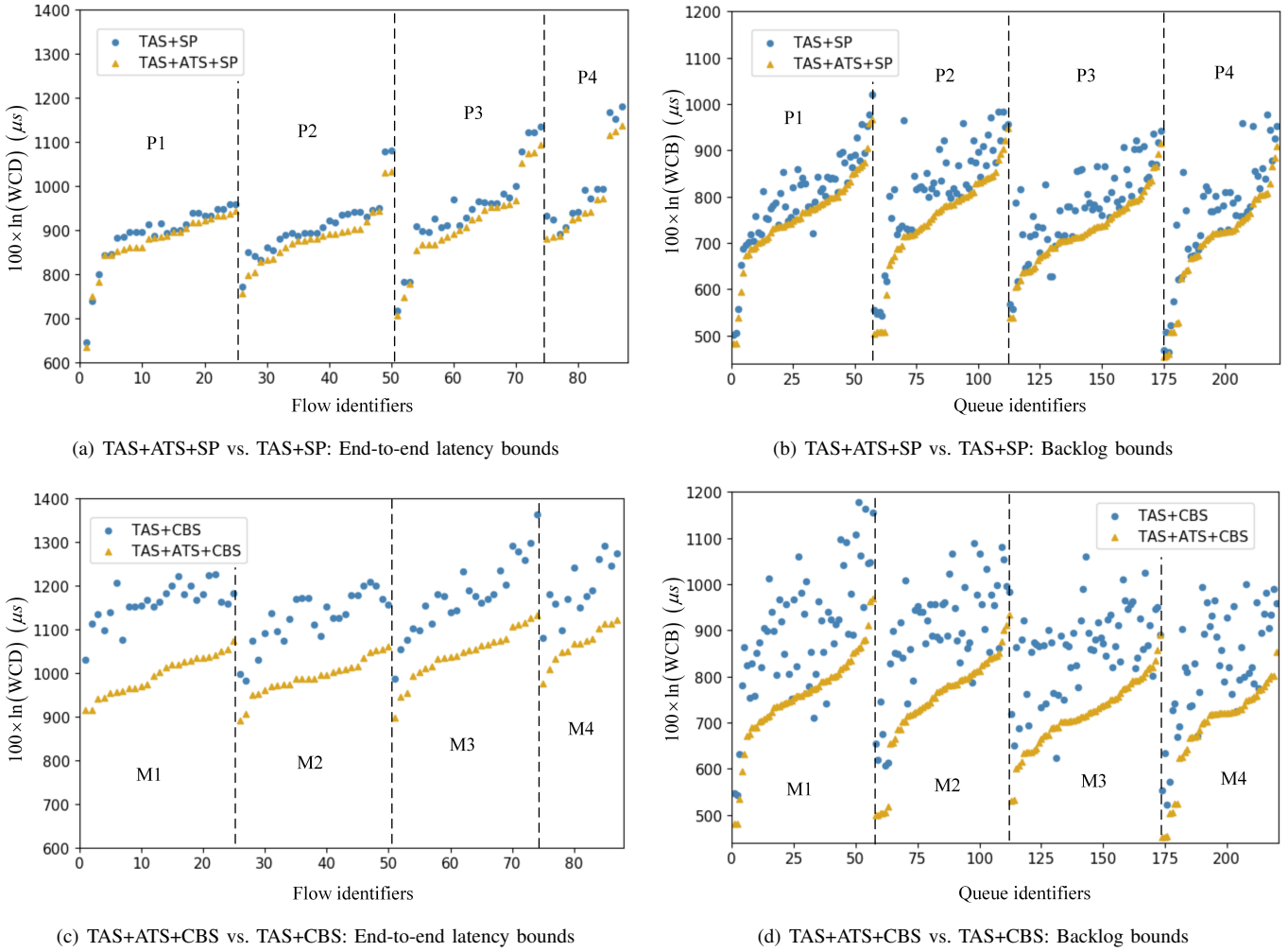
the two novel hybrid architecture of combined traffic shapers TAS+ATS+SP and TAS+ATS+CBS. We have run the NC-based performance analysis method for both combinations TAS+SP and TAS+ATS+SP (resp. TAS+CBS and TAS+ATS+CBS) on the Orion CEV case, and obtained for each combined traffic shaper the upper bounds of the maximum latency (WCD) for each flow and the upper bounds of the maximum backlog (WCB) for each priority queue in an egress port.

We first use the original traffic parameters [45] of the Orion CEV ($\text{TC}_{\text{Orion1}}$), including 99 TT flows and 87 Rate Constraint (RC) flows of the same priority. RC flows are considered into SP and AVB flows under respectively combinations of traffic shapers in this paper. The idle slope for AVB traffic is set to 75%. The average network load (resp. maximum link load) for TT traffic is around 1.5% (resp. 5.5%), and the overall traffic load in the network is 3.5% on average and 10% in maximum. The results are shown in Fig. 16, where the upper bounds are normalized to $100 \times \ln(X)$ with $X = \{\text{WCD}, \text{WCB}\}$. The obtained results are sorted in increasing order by results. As can be found from the figure, even though the backlog upper bounds for most of queues in egress ports under TAS+ATS+SP/AVB perform superior to TAS+SP/AVB, ATS does not show the positive impact on the upper bounds of end-to-end latencies of the flows. This is because that the average traffic load for both TT and SP/AVB is relative low. The results for Orion CEV conform to the outcomes shown in Fig. 13(a), (b) and Fig. 14.

Then we increase the traffic load in Orion CEV by raising the rate and keeping the frame size of the flow (called $\text{TC}_{\text{Orion2}}$). The average network load (resp. maximum link load) for TT traffic is increased to 15% (resp. 54%). The overall traffic load in the network is 25% on average and 69% in maximum. Moreover, we are interested to take a look ATS effect on multiple priorities and thus we classify the SP/AVB traffic into four priorities. There are 25 flows of priority P_1, 25 of priority P_2, 24 of priority P_3 and 13 of priority P_4. For the AVB traffic, due to the increase traffic load and the uneven load for each traffic type on each link, it is difficult to assign a fixed idleSlope for each AVB traffic class across the entire network. Thus, we calculate the idle slope of AVB Class M_i for each egress port according to the actual bandwidth utilization [1, § 8.6.8.2], i.e.,

$$idSl_i = \text{operIdleSlope}(M_i) \cdot \frac{\text{OperCycleTime}}{\text{GateOpenTime}},$$

where $\text{operIdleSlope}(M_i)$ [1, § 34.3] is the actual bandwidth that is currently reserved for the AVB class M_i for each port, and $\frac{\text{OperCycleTime}}{\text{GateOpenTime}}$ is the fraction of effective time that the gate is open for AVB traffic. Similarly, the results are shown in Fig. 17. The obtained results are grouped by different priorities with vertical dotted lines and, respectively, sorted in increasing order by results within each priority. From the figure, we can find that with the increasing traffic load, the combination shaper TAS+ATS+SP (resp. TAS+ATS+CBS) outperforms TAS+SP (resp. TAS+CBS) from the perspective of latency and backlog upper bounds. In addition, since the idle slope of CBS which related to the service ability is set according to the actual bandwidth of AVB traffic, the relative load of AVB is larger



(a) TAS+ATS+SP vs. TAS+SP: End-to-end latency bounds

(b) TAS+ATS+SP vs. TAS+SP: Backlog bounds

(c) TAS+ATS+CBS vs. TAS+CBS: End-to-end latency bounds

(d) TAS+ATS+CBS vs. TAS+CBS: Backlog bounds

Fig. 17. Comparison of combined traffic shapers under the Orion CEV TC_{Orion2} (TT - 15%, SP/AVB with 4 priorities)

than that of SP. Therefore, the positive effect of ATS on CBS in combined TAS+ATS+CBS is more obvious than that on SP in the combined TAS+ATS+SP. What is more interesting is that, with the combination of ATS, the performance of SP and CBS get closing to each other, but CBS allow the bandwidth reservation for the traffic.

VI. Conclusion

This paper has studied the qualitative performance comparison of the various individual traffic shapers and their possible combinations. SP and CBS have as advantages that SP is more beneficial to the transmission delay of high-priority traffic, while CBS can specify bandwidth reservation for each priority traffic. In addition, due to the credit controlling by CBS, the long-term rate of traffic arrival is reduced, thus it is possible that backlog upper bounds of AVB traffic are lower than SP traffic. Compared with SP and CBS, ATS has limited advantages for high-priority traffic. Only when the average traffic load of high-priority traffic in the network reaches around 80%, ATS can show its superiority. The positive effect of ATS on low priority traffic is more obvious. When the average traffic load of low-priority traffic of the entire network reaches about 20% (overall load 40%), the positive effect of ATS on the upper bound of delay performance begins to become prominent. In addition, when the average load of low-priority traffic reaches about 5% (overall load 10%), the effect of ATS on the upper bound of port backlog performance becomes positive.

Compared with all the above traffic shapers, TT traffic implemented with flow-based scheduling by TAS has the highest performance, with ultra low latency, jitter and backlog. However, it is well known that TAS requires the synthesis of optimized GCLs, to which does not scale to large networks with many flows. This problem can be mitigated by combining different traffic shapers in the same switch architecture, to reduce the number of flows handled by TAS. Moreover, the combined use of ATS with TAS will make ATS play a more active role, of which the effect is similar to the reshaping impact of ATS used individually on low priority traffic, and at the same time, TAS will maintain unchanged its advantages of ultra low latency and jitter.

References

[1] IEEE, “802.1Q—IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks,” https://standards.ieee.org/standard/802_1Q-2018.html, 2018.
[2] IEEE, “802.3 Standard for Ethernet,” 2015.
[3] IEEE, “802.1AS-Rev—Timing and Synchronization for Time-Sensitive Applications,” http://www.ieee802.org/1/pages/802.1AS-rev.html, 2016.
[4] IEEE, “802.1Qbv—Enhancements for Scheduled Traffic,” http://www.ieee802.org/1/pages/802.1bv.html, 2016.
[5] IEEE, “802.1Qcr—IEEE Standard for Local and metropolitan area networks - Bridges and Bridged Networks Amendment: Asynchronous Traffic Shaping,” https://1.ieee802.org/tsn/802-1qcr/, 2018.
[6] IEEE, “802.1BA—Audio Video Bridging (AVB) Systems,” http://www.ieee802.org/1/pages/802.1ba.html, 2011.
[7] IEEE, “802.1Qav—Forwarding and Queuing Enhancements for Time-Sensitive Streams,” https://www.ieee802.org/1/pages/802.1av.html, 2009.
[8] IEEE, “802.1ASrev—Timing and Synchronization for Time-Sensitive Applications,”http://www.ieee802.org/1/pages/802.1AS-rev.html, 2017.

[9] IEEE, "802.1Qbu—Frame Preemption," http://www:ieee802:org/1/pages/802.1bu.html, 2015.

[10] S. S. Craciunas, and R. S. Oliver, "An overview of scheduling mechanisms for time-sensitive networks," in *Proc. of the Real-time summer school LÉcole dÉté Temps Réel (ETR)*, 2017.

[11] F. Durr, and N. G. Nayak, "No-wait packet scheduling for IEEE time-sensitive networks (TSN)," in *Proc. of the 24th International Conference on Real-Time Networks and Systems*, 2016.

[12] S. S. Craciunas, R. S. Oliver, M. Chmelik, and W. Steiner, "Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks," in *Proc. of the 24th International Conference on Real-Time Networks and Systems*, ACM, 2016.

[13] P. Pop, M. L. Raagaard, S. S. Craciunas, and W. Steiner, "Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks," IET Cyber-Physical Systems: Theory & Applications, 1(1), 2016.

[14] M. L. Raagaard, "Algorithms for the optimization of safety-critical networks," Doctoral dissertation, Master's thesis, DTU, 2017.

[15] M. Vlk, Z. Hanzálek, K. Brejchová, S. Tang, S. Bhattacharjee, and S. Fu, "Enhancing Schedulability and Throughput of Time-Triggered Traffic in IEEE 802.1 Qbv Time-Sensitive Networks," *IEEE Transactions on Communications*, 2020.

[16] R. S. Oliver, S. S. Craciunas, and W. Steiner, "IEEE 802.1 Qbv gate control list synthesis using array theory encoding," in *Proc. of IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2018.

[17] L. Zhao, P. Paul, and S. S. Craciunas, "Worst-case latency analysis for IEEE 802.1 Qbv time sensitive networks using network calculus," IEEE Access, vol.6, 2018.

[18] L. Zhao, P. Pop, Z. Gong, and B. Fang, "Improving Latency Analysis for Flexible Window-Based GCL Scheduling in TSN Networks by Integration of Consecutive Nodes Offsets," IEEE Internet of Things Journal, 2020.

[19] D. Hellmanns, J. Falk, A. Glavackij, R. Hummen, S. Kehrer, and F. Durr, "On the Performance of Stream-based, Class-based Time-aware Shaping and Frame Preemption in TSN," in *Proc. of IEEE International Conference on Industrial Technology (ICIT)*, 2020.

[20] R. Mahfouzi, A. Aminifar, S. Samii, A. Rezine, P. Eles, and Z. Peng, "Stability-aware integrated routing and scheduling for control applications in Ethernet networks," in *Proc. of Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2018.

[21] J. Specht, and S. Samii, "Urgency-based scheduler for time-sensitive switched ethernet networks," in *Proc. of IEEE Euromicro Conference on Real-Time Systems (ECRTS)*, 2016.

[22] J. Specht, and S. Samii, "Synthesis of queue and priority assignment for asynchronous traffic shaping in switched ethernet," in *Proc. of IEEE Real-Time Systems Symposium (RTSS)*, 2017.

[23] Z. Zhou, Y. Yan, M. Berger, and S. Ruepp, "Analysis and modeling of asynchronous traffic shaping in time sensitive networks," in *Proc. of 14th IEEE International Workshop on Factory Communication Systems*, 2018.

[24] Z. Zhou, M. S. Berger, S. R. Ruepp, and Y. Yan, "Insight into the IEEE 802.1 Qcr asynchronous traffic shaping in time sensitive network," *Advances in Science, Technology and Engineering Systems Journal*, 4(1), pp. 292-301, 2019.

[25] J. Y. Le Boudec, "A theory of traffic regulators for deterministic networks with application to interleaved regulators," *IEEE/ACM Transactions on Networking*, vol.26, no.6, pp. 2721-2733, 2018.

[26] J. Diemer, J. Rox, and R. Ernst, "Modeling of ethernet avb networks for worst-case timing analysis," *IFAC Proceedings Volumes*, vol. 45, no. 2, 2012.

[27] U. D. Bordoloi, A. Aminifar, P. Eles, and Z. Peng, "Schedulability analysis of Ethernet AVB switches," in *IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*, 2014.

[28] J. A. R. De Azua, and M. Boyer, "Complete modelling of AVB in network calculus framework," in *Proc. of the 22nd International Conference on Real-Time Networks and Systems*, 2014.

[29] L. Zhao, F. He, and E. Li, "Improving worst-case delay analysis for traffic of additional stream reservation class in Ethernet-AVB Network," Sensors, vol 18, no. 11, 2018.

[30] J. Cao, P. J. L. Cuijpers, R. J. Bril, and J. J. Lukkien, "Tight worst-case response-time analysis for ethernet AVB using eligible intervals," in *Proc. of IEEE World Conference on Factory Communication Systems (WFCS)*, 2016.

[31] L. L. Bello, "Novel trends in automotive networks: A perspective on Ethernet and the IEEE Audio Video Bridging," in *Proc. of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014.

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

[33] L. Zhao, P. Pop, Z. Zheng, and Q. Li, "Timing analysis of AVB traffic in TSN networks using network calculus," in *Proc. of IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2018.

[34] L. Zhao, P. Pop, Z. Zheng, H. Daigmorte, and M. Boyer, "Latency Analysis of Multiple Classes of AVB Traffic in TSN with Standard Credit Behavior using Network Calculus," *IEEE Transactions on Industrial Electronics*, in press, 2020.

[35] E. Mohammadpour, E. Stai, M. Mohiuddin, and J. Y. Le Boudec, "Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping," in *Proc. of 30th International Teletraffic Congress*, 2018.

[36] B. Fang, L. Qiao, Z. Gong, and H. Xiong, "Simulative Assessments of Credit-Based Shaping and Asynchronous Traffic Shaping in Time-Sensitive Networking," in *Proc. of 12th International Conference on Advanced Infocomm Technology*, pp. 111-118, 2020.

[37] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, 2018.

[38] L. Maile, K. S. Hielscher, and R. German, "Network Calculus Results for TSN: An Introduction," in *Proc. of Information Communication Technologies Conference (ICTC)*, 2020.

[39] N. Reusch, L. Zhao, S. S. Craciunas, and P. Pop, "Window-based schedule synthesis for industrial IEEE 802.1 Qbv TSN networks," in *Proc. of 16th IEEE International Conference on Factory Communication Systems (WFCS)*, pp. 1-4, 2020.

[40] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. Elbakoury, "Performance Comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS)," *IEEE Access*, vol.7, pp. 44165-44181, 2019.

[41] J. Y. Le Boudec, and P. Thiran, "Network calculus: A theory of deterministic queuing systems for the internet," *Springer-Verlag Lecture Notes on Computer Science*, 5th ed., 2001.

[42] Gavriluţ, V. and Pop, P., 2020. Traffic-type Assignment for TSN-based Mixed-criticality Cyber-physical Systems. ACM Transactions on Cyber-Physical Systems, 4(2), pp.1-27.

[43] S. S. Craciunas, and R. S. Oliver, "Combined task-and network-level scheduling for distributed time-triggered systems," *Real-Time Systems*, 52(2), pp. 161-200, 2016.

[44] D. Tamas-Selicean, "Design of mixed-criticality applications on distributed real-time systems," PhD Thesis, Technical University of Denmark, 2014.

[45] M. Paulitsch, E. Schmidt, B. Gstottenbauer, C. Scherrer, H. Kantz, "Time-triggered communication (industrial applications)". Time-triggered communication, pp 121–152, 2011.

APPENDIX
NETWORK CALCULUS THEORY

Network Calculus [41] is a system theory proposed for analyzing performance guarantees in communication networks. By constructing arrival curve and service curve models, the maximum amount of flow data entered into network nodes and the minimum service offered by network nodes can be obtained. Network Calculus is build on min-plus algebra, which includes two basic operators on non-decreasing functions: $\mathcal{F}_{\uparrow} = \{f : \mathbb{R}_+ \to \mathbb{R} | x_1 < x_2 \Rightarrow f(x_1) < f(x_2), x < 0 \Rightarrow f(x) = 0\}$. One is the convolution operator $\otimes$,

$$(f \otimes g)(t) = \inf_{0 \le s \le t} \{f(t-s) + g(s)\},$$

and the other is the deconvolution operator $\oslash$,

$$(f \oslash g)(t) = \sup_{s \ge 0} \{f(t+s) - g(s)\},$$

where $\inf$ means infimum and $\sup$ means supremum.

The arrival and service curves are defined by means of the min-plus convolution. An arrival curve $\alpha(t)$ is a model constraining the arrival process $R(t)$ of a flow, where $R(t)$ represents the input cumulative function counting the total data bits of the flow that has arrived on the network node up to time t. We say that $R(t)$ is constrained by $\alpha(t)$ iff,

$$R(t) \le \inf_{0 \le s \le t} \{R(s) + \alpha(t-s)\} = (R \otimes \alpha)(t). \tag{35}$$

Note that an arrival curve $\alpha(t)$ should be a non-negative wide-increasing function. A typical example of an arrival curve is the "leaky bucket" constraint satisfying $\alpha(t) = b + r \cdot t$ for

$t > 0$ and $\alpha(0) = 0$, with the maximum burst tolerance b and long-term rate r of the flow.

A service curve $\beta(t)$ models the processing capability of the available resource for the network node. Assume that $R^*(t)$ is the departure process, which is the output cumulative function that counts the total data bits of the flow departure from the network node up to time t. There are several definitions for the service curve. We say that the network node offers the min-plus minimal service curve $\beta(t)$ (considered in this paper) for the flow iff

$$R^*(t) \geq \inf_{0 \leq s \leq t} \{R(s) + \beta(t-s)\} = (R \otimes \beta)(t), \tag{36}$$

and offers the strict service curve $\beta(t)$ iff

$$R^*(t + \Delta t) - R^*(t) \geq \beta(\Delta t), \tag{37}$$

during any backlog period $[t, t+\Delta t)$. Note that a service curve $\beta(t)$ should be a non-negative wide-increasing function.

A shaping curve $\sigma(t)$ characterizes the maximum number of bits that are served during a period of time Δt, which means that the departure process $R^*(t)$ from the server is always constrained by the shaping curve. A server offers a shaping curve $\sigma(t)$ iff $\sigma(t)$ could be an arrival curve for all output process $R^*(t)$, i.e.,

$$R^*(t + \Delta t) - R^*(t) \leq \sigma(\Delta t). \tag{38}$$

Note that the shaping curve $\sigma(t)$ from the output can also be an input arrival curve for the subsequent nodes, but its definition is different from the greedy shaper.

Three basic results of network calculus are given as follows. If the flow $R(t)$ constrained by the arrival curve $\alpha(t)$ traverses the network node offering the service curve $\beta(t)$, the latency experienced by the flow in the network node is bounded by the maximum horizontal deviation between the graphs of two curves $\alpha(t)$ and $\beta(t)$,

$$D = \mathrm{h}(\alpha, \beta) = \sup_{s \geq 0} \{\inf \{\tau \geq 0 \mid \alpha(s) \leq \beta(s+\tau)\}\}. \tag{39}$$

The backlog of the flow in the network node is bounded by the maximum vertical deviation between the graphs of two curves $\alpha(t)$ and $\beta(t)$,

$$B = \mathrm{v}(\alpha, \beta) = \sup_{s \geq 0} \{\alpha(s) - \beta(s)\}. \tag{40}$$

The output arrival curve for the output flow $R^*(t)$ is bounded by the arrival curve $\alpha^*(t)$,

$$\alpha^*(t) = \alpha \oslash \beta(t) = \sup_{s \geq 0} \{\alpha(t+s) - \beta(s)\}, \tag{41}$$

With the known latency upper bound of the flow, the output arrival curve of the flow can also be given by,

$$\alpha^*(t) = \alpha(t) \oslash \delta_D(t), \tag{42}$$

where $\delta_D(t)$ is the burst-delay function with $\delta_D^Q(t)$ being the burst-delay function which equals to 0 if $t \leq D$ and $+\infty$ otherwise. The output arrival curve can also be regarded as the input arrival curve of the flow before reaching the next node.