Run-time Per-Class Routing of AVB Flows in In-Vehicle TSN via Composable Delay Analysis

Weijiang Kong, Majid Nabi, Kees Goossens
Department of Electrical Engineering, Eindhoven University of Technology, the Netherlands
w.kong, m.nabi, k.g.w.goossens@tue.nl

***Abstract*—Time-Sensitive Networking (TSN) is a promising solution for the next generation in-vehicle networks. To enable innovations in adaptability, there is a strong trend to integrate TSN with Software-Defined Networking (SDN). Software-defined TSN requires fast routing algorithms that guarantee the Worst-Case End-to-End Delay (WCED) of the Audio Video Bridging (AVB) flows. However, current AVB flow routing algorithms execute WCED analysis for an exponential amount of route candidates, which is not feasible at run-time. Current WCED analysis of the AVB flows depends on the specific flow setup of other traffic classes. When high-priority flows change, low-priority flows must be adjusted as well to maintain deadline guarantees. In this paper, we propose a per-class flow management scheme to enable run-time routing of the AVB flows. Our composable analysis computes a formal WCED bound of the AVB flows. It does not rely on the flow setup of other traffic classes. So, the WCED and routes can be computed for each traffic class independently and in parallel. Based on the analysis, we develop a fast heuristic algorithm to incrementally route the AVB flows with guaranteed deadlines. Our experiments demonstrate that, compared to the existing solutions, the composable analysis is 2.6x faster. The cost is that the WCED bound can be up to 31% higher. When given a similar run-time as existing solutions, the proposed routing algorithm can set up 1.6x more flows. The average time to set up one flow is within 0.14s.**

***Index Terms*—In-Vehicle TSN, SDN, WCED, Routing.**

I. Introduction

Since the proliferation of automotive applications, Time-Sensitive Networking (TSN) [1] based on the Ethernet standards has emerged as a promising solution for high bandwidth in-vehicle networks. TSN divides mixed-criticality traffic into traffic classes: Time-Triggered (TT) traffic for safety-critical control data, Audio Video Bridging (AVB) traffic for Worst-Case End-to-End Delay (WCED) guaranteed streaming data, and Best Effort (BE) traffic for non-real-time data. AVB flows are further divided into two Stream Reservation (SR) classes according to the service requirement: class A (SR-A) for delay-sensitive audio data and class B (SR-B) for high-bandwidth video data. TSN applies Time Aware Shaping (TAS) [2] and Credit Based Shaping (CBS) [3] in addition to the class-based queuing to ensure zero frame losses and bounded latency.

An emerging trend of today's TSN is Software-Defined Networking (SDN) whose basic approach is to simplify switches into forwarding devices and manage them through logically centralized network control [4]. For in-vehicle networks, SDN provides benefits regarding run-time flow setup which is essential for adaptive network management and fault mitigation. Time-Sensitive Software-Defined Networking (TSSDN) [4] and TSNu [5] have demonstrated successful integration of the TSN data plane and the SDN management protocols.

The controllers of software-defined TSN must route AVB flows with end-to-end deadline guarantees. Although routing in real-time SDN is a widely researched topic, AVB flows needs different approaches to consider the delays caused by TAS and CBS. In conventional SDN, the link delay of a flow is usually modeled as a constant weight of the link [6], [7]. Instead, the link delay of an AVB flow under traffic shaping depends on other flows routed on the same link. It also depends on the jitter of frame arrival, which is caused by the variation of blocking scenarios on upstream links. Thus, the WCED of the AVB flows must be computed by specific analyses [8]–[10].

Routing AVB flows with the constrained WCED is difficult because a route change might require the entire network to be re-evaluated [10]. Existing approaches find route candidates for each flow and execute WCED analyses on their combinations for a good solution [11]–[13]. These approaches are overwhelmed by the exponential search space although meta-heuristic algorithms are used to perform efficient exploration. Additionally, current WCED analyses can make tight delay estimations when based on a specific TT flow setup. Once the TT setup changes, the controllers must re-evaluate the WCED of AVB flows and reroute the flows that lose their deadline guarantees. Similarly, when SR-A flows change, the issue that SR-B flows may violate their deadlines must be resolved.

In this work, we propose a per-class traffic management scheme for TSN, which aims at isolating the management logic between traffic classes. It allocates bandwidth to SR classes based on their relative data rate. Then, the flow routes of each SR class are managed independently. To enable per-class management, our contributions are as follows.

- We propose a heuristic routing algorithm to avoid checking the WCED exponential times. It handles the AVB flows one by one. For each flow, the algorithm first prunes the network to consider only feasible links. Then, metrics correlated with the WCED are used to generate route candidates. The algorithm applies WCED analysis on a polynomial number of route candidates to ensure deadline guarantees at last.
- We propose a composable WCED analysis that bounds the blocking from other classes by analyzing traffic shaping, i.e., its WCED bound is independent of the flow setups in other classes. Hence, when high-priority flows change, low-priority flows do not worry about deadline violations. Its WCED computation is simpler and faster than the existing approaches. Also, it allows the routes and the WCED of SR-A and SR-B to be computed independently and in parallel. The cost is that its WCED bound is more pessimistic.

This paper is structured as follows. Section II reviews the related works. Section III introduces our system model. Section IV describes the composable WCED analysis. The routing algorithm is discussed in Section V. We perform evaluation in Section VI and conclude the paper in Section VII.

II. Related Work

This section introduces the related works of the WCED analysis and routing of AVB flows in software-defined TSN.

Several works have studied routing with end-to-end deadlines. Routing with the end-to-end delay constraints is studied in real-time SDN [6] and vehicular SDN [7]. Both works model delay as weights of the links. This assumption does not apply to TSN whose delay requires specific analysis. [14] and [15] studies routing with guaranteed quality of service based on network calculus. But they target networks with different shaping mechanisms than TSN. Thus, the network calculus model and the resulted routing scheme are different. Several routing approaches have been developed for AVB flows in TSN. [16] and [17] use integer linear programming to route the SR-A flows whose WCED can be formulated with linear constraints. But they cannot handle SR-B flows whose WCED must be achieved with complex algorithms (e.g., the algorithm in [10]). GRASP [11]–[13] uses a stochastic search algorithm to optimize the routes of both SR-A and SR-B flows. It requires checking the WCED for an exponential amount of route candidates. So, the execution time is too long to be used at run-time. Instead, our routing scheme supports both SR class A and B. It only checks the WCED for a polynomial number of route candidates to ensure a short execution time.

The delay of AVB flows has been widely studied. [18] computes the delay of the AVB frames assuming periodic arrival. Since it does not consider the arrival jitter, it is not applicable for WCED. The eligible interval analysis [19] proves that the blocking due to higher and lower priority classes is bounded by the traffic shaping instead of the exact traffic context. Although it is composable, it does not consider the arrival jitter for blocking within each SR class, so it is also not applicable for WCED. Several holistic approaches have been developed to analyze the WCED of AVB flows, including network calculus [20], compositional performance analysis [8], trajectory approach [9], and Forward End-to-end Delay Analysis (FA) [10]. All of them are non-composable approaches based on specific flow setups in other traffic classes. Instead, the proposed analysis provides a composable WCED bound.

III. System Model

A network is a directed graph $G \equiv (V, E)$. The nodes V are switches (V_{SW}) and hosts (V_{ES}). $(u, v) \in E$ is the simplex connection from u to v ($u, v \in V$). A bidirectional link is modeled as two opposite edges. For simplicity, we assume uniform links with bandwidth bw and propagation delay pd. γ is the maximum proportion of bandwidth that can be allocated to SR classes. Its default value in the AVB standard is 0.75 [3].

A flow is a 6-tuple $f = (s_f, D_f, c_f, x_f, T_f, dl_f)$, where $s_f \in V$ is the source, $D_f \subseteq V$ the destinations, c_f the transmit time of the maximum-sized frame (interframe gap included), $x_f \in \{A, B\}$ the SR class, T_f the frame ingress period, and dl_f the end-to-end deadline. The set of flows in SR class $x \in \{A, B\}$ is denoted by F^x and, when routes are specified, the set of flows in SR class x routed on link (u,v) is denoted by $F^x_{u,v}$. The maximum transmit time of the class x flows is denoted by $\overline{c^x}$. $\overline{c^{BE}}$ denotes the maximum transmit time of BE flows. To set up a flow, the network controller allocates route $R_f = \{P_{d,f} \subseteq E |\ \forall d \in D_f\}$, which consists of a path (the links over which frames are forwarded) to every destination of the flow. We list the notations in TABLE I for the rest of the paper.

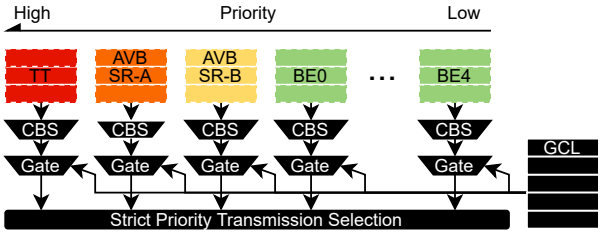


Fig. 1: Traffic queuing and shaping in TSN [2].

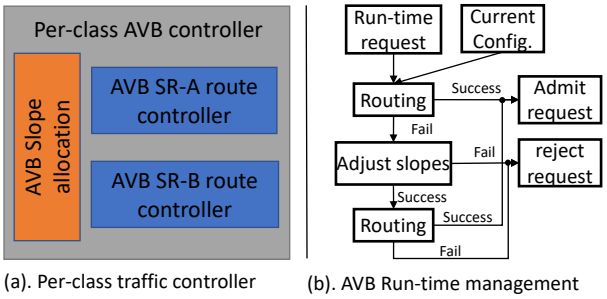
TABLE I: Table of symbols and notations

Network notations			
G	network	pd	propagation delay
V	network nodes	bw	link bandwidth
E	network links	$F^x_{u,v}$	class x flows on (u,v)
γ	(SR) bw limit	t_{ts}	length of time slot
α^x	idle slope	t_{ms}	min service time
β^x	send slope	t_{wb}	max reserved time
Flow (f) notations			
s_f	source	c_f	frame size
D_f	destinations	T_f	transmission period
dl_f	deadline	x_f	SR class
R_f	route	$P_{d,f}$	path to d
$\overline{c^x}$		class x max transmit time	
WCED analysis notations			
$J_{v,f}$	jitter	$Smax_{v,f}$	longest traversal time
$L_{u,v,f}$	max link delay	$Smin_{v,f}$	shortest traversal time
MC^x	max credit	$in(u)$	ingress nodes of u
$W_{u,v,f}(t)$	blocking time	t	frame arrival time
$W^{tas}_{u,v,f}(t)$	TAS blocking	$rbf_{u,f}(t)$	request bound function
$W^{cbs}_{u,v,f}(t)$	CBS blocking	$ibf_{u,f}(t)$	ingress bound function
$W^{other}_{u,v,f}(t)$		blocking by BE and other SR classes	
$W^{same}_{u,v,f}(t)$		blocking from the same SR class	
$Q^x_{w,u,v}$		$\{g \in F^x \|(w,u),(u,v) \in R_g\}$	
$I^x_{u,v}$		$\{g \in F^x \| s_g = u\}$	
$O(WA)$		complexity of WCED analysis	
Routing notations			
k	order of paths	$w_{u,v}$	weight of the links

A. Traffic Shaping

TSN queuing is conducted on a per-class basis. Each class (TT, SR-A, SR-B) has a queue on every egress port shown in Fig. 1. Traffic shaping determines which queue can be transmitted. Several integration policies have been proposed to integrate CBS with a TAS scheduling. This paper assumes the integration policies in TSSDN [21] which offers deterministic delay and rapid run-time configuration.

TAS consists of gates that cyclically open and close controlled by a predefined schedule stored in the gate control list. In TSSDN, the period of the schedule is uniformly divided into time slots with length t_{ts}. The longest duration TT frames can reserve in any time slot is t_{wb} (including guard band and



(a). Per-class traffic controller (b). AVB Run-time management

Fig. 2: Per-class traffic management scheme of TSN

interframe gap). For the rest of the time slot, TAS gates are opened for both AVB and BE traffic. Thus, the *minimum service duration* for AVB and BE traffic is $t_{ms} = t_{ts} - t_{wb}$ in every time slot. Note that t_{ts} and t_{wb} are defined before the network starts and do not change on the run [21].

CBS associates the queue of each SR class with a credit. Frames can start transmission only when the queue holds non-negative credit. In CBS, the desired bandwidth reserved for SR class x is represented by the *idle slope* α^x and the *send slope* is defined as $\beta^x = bw - \alpha^x$. While a queue is transmitting, its credit is decreased at the rate of the send slope. If the (non-empty) queue for SR class x is blocked due to negative credit or other ongoing transmissions, its credit is replenished at the rate of the idle slope. The credit stays constant when the TAS gate is closed and it is set to zero once the queue is empty. For each SR class, we allocate the same send and idle slopes for every link like most of the AVB routing approaches [11], [12], [16], [17], because we focus on finding a good solution within short run-time. Otherwise, searching for idle slopes on every link together with the flow routes can be too complex.

B. Per-Class Management Scheme

In conventional TSSDN where non-composable analyses are used to check the WCED, the problem that changing the high-priority flows may cause the deadline violation of low-priority flows must be solved. Per-class traffic management avoids this issue by isolating the management logic between traffic classes via composable WCED analysis. A per-class traffic controller is shown in Fig. 2(a), which consists of three components: a slope allocation unit that computes the CBS slopes for both SR classes and two independent route controllers for SR-A and SR-B. Since the WCED of the TT traffic is guaranteed via TAS and cannot be disrupted by the AVB and BE traffic, the TT traffic can use the existing management approaches for TSSDN [21].

Both route controllers execute our proposed routing algorithm. First, the network is pruned to consider only feasible links. Then, route candidates are generated based on metrics correlated with the WCED. The algorithm only applies WCED analysis on the generated route candidates, of which the number is polynomial. Because the WCED analysis is composable, routing can be performed independently for each SR class. Here we describe our management scheme in detail.

Initialization: When the network starts, the controller receives the initial set of flows. First, the slope allocation unit partitions the bandwidth between SR-A and SR-B based on their data rate ratio according to Eq. 1. Then, flows are incrementally routed into the network using the proposed routing algorithm. Flows routed successfully can be admitted while the flow requests are rejected if the algorithm fails to find a route.

$$\forall x \in \{A, B\}, \alpha^x = \frac{\gamma \cdot bw \cdot \sum_{f \in F^x} c_f/T_f}{\sum_{y \in \{A,B\}} \sum_{f \in F^y} c_f/T_f} \tag{1}$$

Run-time management: When receiving flow requests, the controller follows the procedure in Fig. 2(b) to maintain the running flows while adding new flows. It handles flow requests one by one. First, routing is attempted without changing the CBS slopes. If it fails, the idle slopes are recalculated by Eq. 1 counting in both the running flows and the flow request. The WCED analysis checks that all running flows still meet their deadlines with the adjusted slopes. Otherwise, the slope adjustment fails and the flow request is rejected. Then, routing is re-attempted with the adjusted slopes.

IV. Composable WCED Analysis

Our composable WCED analysis is presented in this section. It independently establishes the WCED bounds for each SR class. Its jitter computation is based on FA [10]. So we first introduce FA and then present our analysis.

A. Introduction to Forward End-to-end Delay Analysis

The queuing of AVB flows causes blocking over their routes, where blocking on a link causes arrival jitter on downstream links. FA propagates a busy period analysis along the flow routes. On each link $(u, v) \in R_f$, the *shortest and longest traversal time* of flow f from s_f to v, denoted by $Smin_{v,f}$ and $Smax_{v,f}$, are iteratively computed by Eq. 2 [10], in which $L_{u,v,f}$ is the maximum delay of f on (u, v) due to queuing and transmission. $Smin_{s_f,f}$ and $Smax_{s_f,f}$ are zero by definition.

$$\begin{aligned} Smin_{v,f} &= Smin_{u,f} + c_f + pd \\ Smax_{v,f} &= Smax_{u,f} + L_{u,v,f} + pd \end{aligned} \tag{2}$$

The *jitter* is *the maximum difference between the shortest and longest traversal time*, defined as $J_{v,f} = Smax_{v,f} - Smin_{v,f}$. The deadline of f is met if $Smax_{d,f} \leqslant dl_f, \forall d \in D_f$. On each link, FA calculates $L_{u,v,f}$ through a busy period analysis when the arrival jitters are known. Assume a frame of f arrives at time t since the start of the busy period, the *maximum blocking time* $W_{u,v,f}(t)$ is defined as *the maximum time that the frame of f which arrives at time t finishes transmission* (measured since the start of the busy period). $L_{u,v,f}$ is defined by Eq. 3 [10].

$$L_{u,v,f} = \max_{t \geqslant 0} \{W_{u,v,f}(t) - t\} \tag{3}$$

B. Composable Analysis

The main difference between our analysis and FA is that we acquire $W_{u,v,f}(t)$ in a composable manner, i.e., our $W_{u,v,f}(t)$ bound only depends on the flows in class x_f and the traffic shaping. Note that although the sizes of the frames are determined by their applications, it is recommended to have them limited [22]. Thus, $\overline{c^x}$ can be known before the network starts.

$W_{u,v,f}(t)$ can be analyzed by Eq. 4, in which $W^{other}_{u,v,f}$ is the blocking (time) by the transmission of BE and other SR classes,

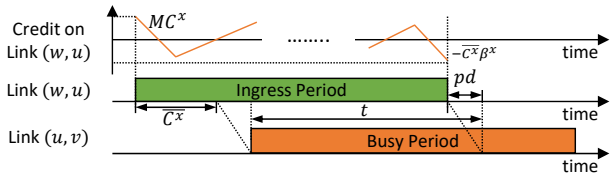


Fig. 3: Worst-case ingression of the upstream CBS

$W^{same}_{u,v,f}$ the blocking by the transmission of the same SR class, $W^{cbs}_{u,v,f}$ the blocking by CBS, and $W^{tas}_{u,v,f}$ the blocking by TAS.

$$W_{u,v,f}(t) = W^{other}_{u,v,f} + W^{same}_{u,v,f}(t) + W^{cbs}_{u,v,f}(t) + W^{tas}_{u,v,f}(t) \quad (4)$$

Blocking from BE and other SR classes: The maximum credit of each SR class is denoted by MC^A and MC^B. The eligible interval analysis proves that the blocking caused by BE and other SR classes is bounded by Eq. 5 [19]. This bound is composable since it is independent of other traffic classes. So we use it directly. Note that since every link has identical idle slopes, their maximum credits are the same as well.

$$\begin{aligned} MC^A &= \alpha^A \cdot (\max\{\overline{c^B}, \overline{c^{BE}}\} \\ MC^B &= \alpha^B \cdot (\overline{c^{BE}}(1 + \frac{\alpha^A}{\beta^A}) + \overline{c^A}) \\ W^{other}_{u,v,f} &= \frac{MC^{x_f}}{\alpha^{x_f}} \end{aligned} \quad (5)$$

Blocking from the same SR class: Two major facts limit the blocking between the flows of an SR class. First, the frame arrival follows the jittered periodic pattern, in which the worst-case scenario happens when the first frame arrives with the maximum jitter and the later frames arrive as early as possible. Thus, the blocking of each flow can be bounded by the *request bound function* ($rbf_{u,g}(t)$) defined in Eq. 6.

$$\forall g \in F^x_{u,v},\ rbf_{u,g}(t) = (1 + \left\lfloor \frac{t + J_{u,g}}{T_g} \right\rfloor) c_g \quad (6)$$

Second, frames arrived from the same upstream link are limited by its CBS. This is referred to as the serialization effect [9], [10]. However, existing works address the serialization effect in a non-composable manner, i.e., the SR-A flows must be known to analyze SR-B flows. Our approach solves this issue. To consider the serialized flows, we define $in(u) = \{w \in V | (w,u) \in E\}$ and distinguish flows based on where they are ingressed from as Eq. 7.

$$\begin{aligned} Q^x_{w,u,v} &= \{g \in F^x | (w,u),(u,v) \in R_g\}, \forall w \in in(u) \\ I^x_{u,v} &= \{g \in F^x | s_g = u\} \end{aligned} \quad (7)$$

The worst-case ingress scenario on the upstream link is illustrated in Fig. 3. During $[0,t]$, the first arrived frame may be maximum-sized while the last frame can have an arbitrary size. So the duration in which frames transmitted on (w,u) can reach u is bounded by $t + \overline{c^x}$. The CBS holds maximum credit before the ingress period. When the ingress period ends, minimum credit is left. Hence, the frame arrival is bounded by the *ingress bound function* ($ibf_{w,u,f}(t)$) defined in Eq. 8.

$$\begin{aligned} &\alpha^{x_f}(t + \overline{c^x} - ibf_{w,u,f}(t)) - \beta^{x_f} ibf_{w,u,f}(t) \\ &\quad \geqslant -MC^{x_f} - \beta^{x_f}\overline{c^x} \\ &\Rightarrow ibf_{w,u,f}(t) \leqslant \frac{\alpha^{x_f}}{bw} \cdot t + \frac{MC^{x_f}}{bw} + \overline{c^x} \end{aligned} \quad (8)$$

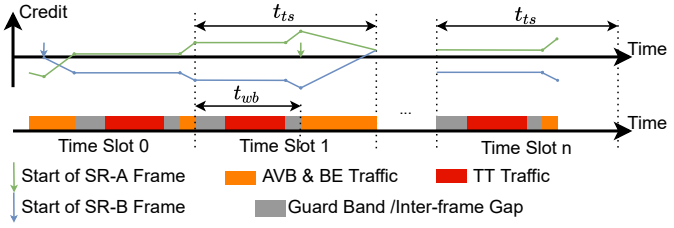


Fig. 4: Worst-case blocking scenarios due to TAS

$W^{same}_{u,v,f}(t)$ bounded both by the request bound and the ingress bound can be calculated by Eq. 9.

$$\begin{aligned} W^{same}_{u,v,f}(t) = &\sum_{g \in I^{x_f}_{u,v}} rbf_{u,g}(t) + \\ &\sum_{w \in in(u)} \min\{\sum_{g \in Q^{x_f}_{w,u,v}} rbf_{u,g}(t), ibf_{w,u,f}(t)\} \end{aligned} \quad (9)$$

CBS Blocking: A composable bound of $W^{cbs}_{u,v,f}(t)$ can be achieved by relaxing the bound in FA, which is in Eq. 10.

$$W^{cbs}_{u,v,f}(t) = \max\{0, W^{same}_{u,v,f}(t) - c_f\} \cdot \frac{\beta^{x_f}}{\alpha^{x_f}} \quad (10)$$

TAS Blocking: The blocking due to TAS occurs during $[0, W_{u,v,f}(t)]$. In the worst-case, two consecutive durations are reserved before the AVB and BE flows can transmit, which is illustrated in Fig. 4. Thus, $W_{u,v,f}(t)$ and $W^{tas}_{u,v,f}(t)$ are correlated as shown by Eq. 11.

$$W^{tas}_{u,v,f}(t) = (1 + \left\lceil \frac{W_{u,v,f}(t)}{t_{ts}} \right\rceil) \cdot t_{wb} \quad (11)$$

Combining Eq. 4 and 10, $W_{u,v,f}(t)$ can be solved by eliminating $W^{tas}_{u,v,f}(t)$ as Eq. 12 shows.

$$\begin{aligned} &W^{other}_{u,v,f} + W^{same}_{u,v,f}(t) + W^{cbs}_{u,v,f}(t) \\ &\quad = W_{u,v,f}(t) - (1 + \left\lceil \frac{W_{u,v,f}(t)}{t_{ts}} \right\rceil) \cdot t_{wb} \\ &\Rightarrow W_{u,v,f}(t) \leqslant \frac{2t_{wb}t_{ts}}{t_{ms}} \\ &\quad + (W^{other}_{u,v,f} + W^{same}_{u,v,f}(t) + W^{cbs}_{u,v,f}(t)) \frac{t_{ts}}{t_{ms}} \end{aligned} \quad (12)$$

To conclude, given a time t, the maximum blocking time can be computed by Eq. 12, in which $W^{other}_{u,v,f}$ can be computed by Eq. 5, $W^{same}_{u,v,f}(t)$ by Eq. 9, and $W^{cbs}_{u,v,f}(t)$ by Eq. 10. To find the maximum delay $L_{u,v,f}$, FA checks t with a constant step (e.g., 1µs) starting from 0. It can stop once $W_{u,v,f}(t) \leqslant t$ which indicates that the busy period ends. Theorem 1 provides a sufficient condition for the stop condition to occur. A possible consequence of violating this condition is infinite backlog.

Theorem 1. *A sufficient condition for the t to exist so that $W_{u,v,f}(t) \leqslant t$ is that $\sum_{g \in F^{x_f}_{u,v}} \frac{c_f}{T_f} < \frac{t_{ms}\alpha^{x_f}}{t_{ts}bw}$.*

Proof. The request bound function has an upper bound $rbf_{u,g}(t) \leqslant \frac{c_g}{T_g}t + \frac{T_g + J_{u,g}}{Tg} c_f$. Hence, $W^{same}_{u,v,f}(t)$ in Eq. 9 can be upper bounded by the sum of these bounds for all $g \in F^{x_f}_{u,v}$, which is also a linear function of t with coefficient $\sum_{g \in F^{x_f}_{u,v}} \frac{c_g}{T_g}$. When t is sufficiently large so that $W^{same}_{u,v,f}(t) \geqslant c_f$, the

$W^{cbs}_{u,v,f}(t)$ in Eq. 10 is linear regarding $W^{same}_{u,v,f}(t)$ with the coefficient $\frac{\beta^{x_f}}{\alpha^{x_f}}$. In such scenarios, $W_{u,v,f}(t)$ in Eq. 5 can be upper bounded by a linear function of t whose coefficient is $\frac{bw \times t_{ts} \times \sum_{g \in F^{x_f}_{u,v}} \frac{c_g}{T_g}}{t_{ms} \times \alpha^{x_f}}$. A sufficient condition for the t to exist so that $W^{same}_{u,v,f}(t) \leqslant t$ is that the coefficient of the upper bound of $W^{same}_{u,v,f}(t)$ is less than one, i.e., $\sum_{g \in F^{x_f}_{u,v}} \frac{c_f}{T_f} < \frac{t_{ms}\alpha^{x_f}}{t_{ts} bw}$. □

V. WCED-Aware Incremental Routing

The WCED analysis evaluates all flows in an SR class through the entire network. Since the link delay depends on the arrival jitters, assigning weights whose sum over route bounds the WCED is not feasible. Existing approaches [11]–[13] optimize the routes of all flows at the same time, whose search space is exponential. To find a solution within the limited time, we route the flows incrementally, i.e., flows are added one by one to the traffic so that neither the added flow nor the existing flows violate the deadline. But even for individual flows, the number of route candidates can be exponential. So we propose a heuristic algorithm to generate a polynomial number of route candidates for WCED checking.

Algorithm 1 aims at finding a solution by running the WCED analysis polynomial times. First, it prunes the networks to enforce the sufficient condition in Theorem 1 to avoid infinite backlog (line 3). Then, metrics correlated with the WCED are used as weights of the links to generate a polynomial number of route candidates. Finally, it applies the proposed WCED analysis to these route candidates to ensure deadline guarantee.

Route candidates can be generated in two steps. The first finds path candidates to each destination and the second combines them together. To find path candidates, Algorithm 1 assigns a weight to each link based on metrics correlated with the WCED (line 5) and applies *k* shortest path algorithm [23] (line 8-10). There are several options to set the weights, which we compare experimentally later.

- *Hop-based:* the weight of each link is 1 since routes with fewer hops potentially suffer less blocking.
- *Utilization-based:* the weight of the link (u, v) is set as $\sum_{g \in F^{x_f}_{u,v}} c_g/T_g$ to avoid heavily utilized links.
- *Delay-based:* the weight of the link is set as Eq. 13 to avoid interfering flows with high WCED.

$$w_{u,v} = \max_{g \in F^{x_f}_{u,v}} \max_{d \in D_g} \{\frac{Smax_{d,g}}{dl_g}\} \tag{13}$$

Then, Algorithm 1 combines the path candidates into route candidates (line 11-22). Since *k* path candidates are computed for each destination, they can form $k^{|D_f|}$ route candidates. Checking all of them is still expensive. Hence, we generate route candidates heuristically based on the upper bound of the sum of weights in Eq. 14. This bound is valid for positive weights according to the cardinality of the set union.

$$\sum_{(u,v) \in R_f} w_{u,v} \leqslant \sum_{(u,v) \in P_{d,f}} w_{u,v} + \sum_{d' \in D_f \setminus \{d\}} \sum_{(u,v) \in P_{d',f} \setminus P_{d,f}} w_{u,v} \tag{14}$$

Instead of minimizing the sum of weights, this bound is used as an alternative objective. When a path candidate $P_{d,f}$ is formed into route candidates, the bound is minimized if the paths to other destinations $d' \in \{D_f \setminus d\}$ have a minimum value of $\sum_{(u,v) \in P_{d',f} \setminus P_{d,f}} w_{u,v}$. Such paths can be selected in *O(k|V|)* (*GetMinWeightBound* in line 17). Algorithm 1 constructs one route candidate based on every path candidate by minimizing the alternative objective, resulting in $k \times |D_f|$ route candidates. They are increasingly sorted in the *route candidate queue* (*RCQ*, line 20) based on the sum of weights. At last, the candidates in *RCQ* are checked for WCED. The first one which satisfies the deadline of all flows is returned (line 23-29).

Algorithm 1: AVB incremental routing

```
   input : Network G; existing flows F; added flow f
   output: Route of the added flow R_f
   /* construct path candidate                    */
1  for (u, v) ∈ E do
2      if c_f/T_f + Σ_{g∈F^{x_f}_{u,v}} c_g/T_g > t_ms α^{x_f} / (t_ts bw) then
3          G.RemoveEdge(u, v)
4      else
5          G.SetWeights(u, v, F)
6      end
7  end
8  for d ∈ D_f do
9      K_d = KShortestPath(G, s_f, d)
10 end
   /* construct route candidate                   */
11 RCQ = ∅ // Route candidate queue
12 for d ∈ D_f do
13     for P_d ∈ K_d do
14         RC = ∅ // Route candidate
15         RC.SetPath(d, P_d)
16         for d' ∈ D_f \ {d} do
17             P_{d'} = GetMinWeightBound(K_{d'}, P_d)
18             RC.SetPath(d', P_{d'})
19         end
20         RCQ.Insert(RC)
21     end
22 end
   /* verify route candidate (WCED)               */
23 for R ∈ RCQ do
24     f.SetRoute(R)
25     ComputeWCED(G, F ∪ {f})
26     if ∀g ∈ F^{x_f}, ∀d ∈ D_g, Smax_{d,g} ⩽ dl_g then
27         return R
28     end
29 end
```

The complexity of the proposed WCED analysis is pseudo-polynomial similar to the FA [10]. Assume it is denoted by *O(WA)*. Algorithm 1 takes $|V|^2(|F|+k|V|)$ to compute weights and *k* shortest paths, $k^2|D_f|^2|V|$ to construct route candidates, and $k|D_f|O(WA)$ to check the route candidates. Thus, Algorithm 1 has pseudo-polynomial complexity as well.

VI. Evaluation

To evaluate the proposed approach, we use two industrial topologies: the architecture of the ORION crew exploration vehicle [24] and an industrial automation case study

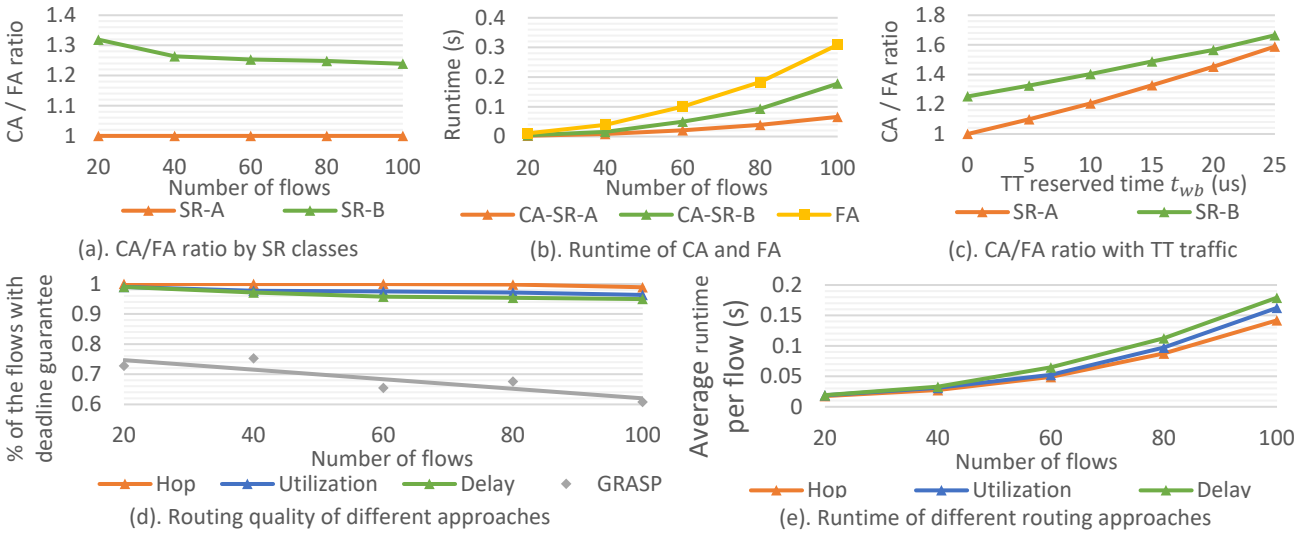


Fig. 5: Result of the WCED analysis and routing evaluation.

from ABB [12]. Both networks consist of uniform links (bw=100Mbps, pd=5.21µs). γ is set to 0.75 as the default value in the AVB standard [3]. Because the original use cases do not have SR-B flows, we generate test cases with randomized flows. Every test case has the same number of SR-A and SR-B flows. Every flow has one source and two destinations selected with uniform distribution. Frames are periodically ingressed from the sources whose specifications follow the latest AVB profile [22] shown in TABLE II. Our Python-based implementation runs on an Intel Core i7-7700K processor.

TABLE II: The flow template adopted from [22]

SR Class	T_f (µs)	c_f (µs)	DL_f (ms)
SR-A	125	9.28	2
SR-B	1333.33	87.2	15

A. WCED Analysis Evaluation

We compare our composable delay analysis (CA) and FA [10] in terms of tightness and run-time. We generate test cases with 20, 40, 60, 80, 100 flows. For each number of flows, 10 random test cases are generated on each topology resulting $2 \times 5 \times 10 = 100$ test cases in total. The flow routes are provided using our heuristic algorithm with hop-based weight. Both CA and FA check the busy period with the step of 10µs. The quality of CA is evaluated by the *CA/FA ratio* which is *the ratio between the WCED computed by CA and FA*.

Since FA originally does not consider TT blocking, we first compare CA and FA without TT flows, i.e., blocking due to TAS is always zero. Fig. 5(a) shows the average CA/FA ratio versus the number of flows. Without any higher priority traffic, CA and FA produce the same results for SR-A, i.e., the CA/FA is always one. For SR-B instead, the average CA/FA ratio is up to 1.31 (when the networks have 20 flows). The cause of the pessimism is that CA only considers the worst-case CBS behavior to bound the WCED so that the WCED is valid regardless of the SR-A flows. Instead, FA can consider the worst-case ingression behavior of the SR-A flow setup to reduce the WCED bound of SR-B flows. As the network load increases, the worst-case ingression approaches the worst-case CBS behavior. Thus, the pessimism reduces, e.g., the average C/F ratio is reduced to 1.23 in test cases with 100 flows.

Fig. 5(b) shows the average run-time of CA and FA. CA allows SR-A and SR-B flows to be analyzed independently. So their run-times are separately measured and the largest one is the run-time for ideal parallel execution. FA instead requires the ingress jitter of SR-A to analyze SR-B. So, SR-A and SR-B must be analyzed in series. The run-times of both CA and FA increases with the network load mainly because the busy period to be checked becomes longer. Similarly, the busy periods of SR-A flows are typically shorter than SR-B flows so CA is faster in analyzing the SR-A flows. CA simplifies the calculation of blocking due to higher-priority classes. Thus, it is 1.6x faster than FA even without parallelism: CA takes 0.023s (0.008s for SR-A, 0.015s for SR-B) in total while FA takes 0.039s in test cases with 40 flows. With ideal parallelism, the speedup is 0.039/0.015=2.6x for the same test cases.

To evaluate the quality of CA with TT flows, we generate 10 test cases with 60 flows on each topology resulting in 20 test cases. For each test case, we set t_{ts} to 200µs and vary t_{wb} from 0 (no TT flow) to 25µs. CA is applied to analyze the WCED considering TAS. Meanwhile, FA analyzes the WCED without TT blocking since it does not consider TAS. The resulting CA/FA ratio versus t_{wb} is shown in Fig. 5(c). For both SR-A and SR-B flows, the CA/FA ratio is approximately linear regarding t_{wb} as Eq. 12 is based on a linear upper bound of the TAS blocking. The increment of WCED is more significant in SR-A flows than in SR-B. For instance, the CA/FA ratios of SR-A vary from 1 to 1.59 while those of SR-B vary from 1.25 to 1.66. The main reason is that the jitter of the SR-B flows is initially higher due to the blocking of SR-A flows. So, while adding TT flows further increases the jitter, its impact on WCED is less significant for SR-B flows than SR-A.

B. Routing Evaluation

We compare the proposed routing algorithm using different weight options (hop, utilization, delay) with the GRASP algorithm [12], which is a state-of-the-art AVB routing algorithm

supporting arbitrary SR classes. Similarly, we generate random test cases with 20, 40, 60, 80, 100 flows, 10 test cases on each topology for every number of flows (100 in total). The run-time of GRASP depends on a user-specified timeout and the original setting is 15 minutes [12] which is impossible for run-time flow setup. Thus, for each test case, we set the timeout of GRASP to the average run-time achieved by running our algorithm with the three weight options, i.e., we keep the run-time of GRASP similar to our algorithm and compare *the number of flows successfully routed with deadline guarantees*. Both algorithms use CA to compute WCED. Note that CA allows SR-A and SR-B flows to be routed in parallel. But similar parallelism can be implemented for GRASP if it uses CA as well. Thus, the comparison is performed without parallel execution for better fairness. The order of the shortest path k is set to 10 as the minimum value resulting in maximum routability.

The average percentage of the flows with deadline guarantees versus the number of flows is shown in Fig. 5(d). As network load increases, more blocking occurs between the frames. So the percentage of the flows with deadline guarantees reduces. Since GRASP performs stochastic search, the results scatter around the trend line. Given a similar amount of time, our algorithm routes 1.6x more flows than GRASP, e.g., it successfully routes 99% of the flows in test cases with 100 flows while GRASP only routes 61% of the flows. The differences caused by the weight options are within 5%. Among the three options tested, hop-based weight leads to the best performance. It is because routing flows with fewer hops reduces the overall network utilization so more flows can be added later. It also indicates that, in networks with uniform links and load, routing flows with fewer hops can reduce WCED more efficiently than routing flows on less crowded links.

The per-flow average run-time of our algorithm with different weight options is shown in Fig. 5(e). The average run-time increases when there are more flows in the network for two reasons. First, when the network load increases, the busy period becomes longer. Thus, the WCED computation becomes slower. Second, a higher network load potentially leads to fewer route candidates satisfying flow deadlines. So, more route candidates need to be checked. The run-time shows a similar trend with the routing quality, i.e., weight options that can set up more flows have shorter run-time. Among the three options, hop-based weight leads to the best run-time, e.g., its average per-flow run-time is 0.14s in test cases with 100 flows, because it potentially reduces the overall network utilization.

VII. Conclusion

In this paper, we propose a per-class flow management scheme for software-defined in-vehicle TSN. To enable per-class flow management, we introduce composability into the WCED analysis of AVB flows. The composable WCED bound is valid regardless of the changes in other traffic classes. So, when high-priority flows change, low-priority flows do not worry about the deadline violation. It also improves the analysis time by up to 2.6x by simplifying the computation and enabling parallelism. The cost in exchange is that the WCED is more pessimistic for SR-B flows as well as SR-A flows with the presence of TT flows. Based on the composable analysis, we propose a heuristic routing algorithm that guarantees the deadline of the AVB flows. It generates route candidates based on the link weights correlated to WCED and applies the composable WCED analysis to ensure deadline guarantees. The experiments show that our algorithm with hop-based weights can route up to 1.6x more flows than existing solutions when a similar amount of run-time is consumed.

References

[1] *Time-Sensitive Networking (TSN) Task Group*, (accessed Sep. 3, 2021). [Online]. Available: https://1.ieee802.org/tsn/#TSN_Standards

[2] "IEEE standard for local and metropolitan area networks–amendment 25: Enhancements for scheduled traffic," *IEEE Std 802.1Qbv*, pp. 1–57, 2016.

[3] "IEEE standard for local and metropolitan area networks–audio video bridging (AVB) systems," *IEEE Std 802.1BA*, pp. 1–45, 2011.

[4] T. Hackel, P. Meyer, F. Korf, and T. C. Schmidt, "Software-defined networks supporting time-sensitive in-vehicular communication," in *VTC2019-Spring*, 2019, pp. 1–5.

[5] V. Balasubramanian, M. Aloqaily, and M. Reisslein, "An SDN architecture for time sensitive industrial IoT," *Computer Networks*, 2021.

[6] R. Kumar, M. Hasan, S. Padhy, K. Evchenko, L. Piramanayagam, S. Mohan, and R. B. Bobba, "End-to-end network delay guarantees for real-time systems using SDN," in *RTSS2017*, 2017, pp. 231–242.

[7] O. Sadio, I. Ngom, and C. Lishou, "Design and prototyping of a software defined vehicular networking," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 842–850, 2020.

[8] J. Diemer, D. Thiele, and R. Ernst, "Formal worst-case timing analysis of Ethernet topologies with strict-priority and AVB switching," in *SIES2012*, 2012, pp. 1–10.

[9] X. Li and L. George, "Deterministic delay analysis of AVB switched Ethernet networks using an extended trajectory approach," *Real-Time Syst.*, vol. 53, no. 1, p. 121–186, 2017.

[10] N. Benammar, H. Bauer, F. Ridouard, and P. Richard, "Timing analysis of AVB Ethernet network using the forward end-to-end delay analysis," in *RTNS2018*, 2018, p. 223–233.

[11] P. Pop, M. Raagaard, S. Craciunas, and W. Steiner, "Design optimization of cyber-physical distributed systems using IEEE time-sensitive networks," *IET Cyber-Phys. Syst: Theory & Appl.*, vol. 1, 11 2016.

[12] S. M. Laursen, P. Pop, and W. Steiner, "Routing optimization of AVB streams in TSN networks," *SIGBED Rev.*, vol. 13, no. 4, p. 43–48, 2016.

[13] V. Gavriluţ, L. Zhao, M. L. Raagaard, and P. Pop, "AVB-aware routing and scheduling of time-triggered traffic for TSN," *IEEE Access*, 2018.

[14] J. W. Guck, M. Reisslein, and W. Kellerer, "Function split between delay-constrained routing and resource allocation for centrally managed QoS in industrial networks," *IEEE Trans. Industr. Inform.*, vol. 12, 2016.

[15] S. Zhu, Z. Sun, Y. Lu, L. Zhang, Y. Wei, and G. Min, "Centralized QoS routing using network calculus for SDN-based streaming media networks," *IEEE Access*, vol. 7, pp. 146 566–146 576, 2019.

[16] A. Atallah, G. Hamad, and O. Mohamed, "Reliability-aware routing of AVB streams in TSN networks," in *Recent Trends and Future Technology in Applied Intelligence*, 2018, pp. 697–708.

[17] ——, "Multipath routing of mixed-critical traffic in time sensitive networks," in *Recent Trends and Future Technology in Applied Intelligence*, 2019, pp. 504–515.

[18] U. D. Bordoloi, A. Aminifar, P. Eles, and Z. Peng, "Schedulability analysis of Ethernet AVB switches," in *RTCSA2014*, 2014, pp. 1–10.

[19] J. Cao, P. J. Cuijpers, R. J. Bril, and J. J. Lukkien, "Independent yet tight WCRT analysis for individual priority classes in Ethernet AVB," in *RTNS2016*, 2016, p. 55–64.

[20] J. A. R. De Azua and M. Boyer, "Complete modelling of AVB in network calculus framework," in *RTNS2014*, 2014, p. 55–64.

[21] N. G. Nayak, F. Dürr, and K. Rothermel, "Incremental flow scheduling and routing in time-sensitive software-defined networks," *IEEE Trans. Industr. Inform.*, vol. 14, no. 5, pp. 2066–2075, May 2018.

[22] "Automotive Ethernet AVB functional and interoperability specification," *AVnu Alliance Whitepaper*, Nov 2019.

[23] J. Y. Yen, "Finding the K shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[24] D. Tămaş-Selicean, P. Pop, and W. Steiner, "Design optimization of TTEthernet-based distributed real-time systems," *Real-Time Syst.*, vol. 51, no. 1, p. 1–35, Jan. 2015.