SynAVB: Route and Slope Synthesis Ensuring Guaranteed Service in Ethernet AVB

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—**In the automotive domain, Ethernet Audio Video Bridging (AVB) provides service guarantees for real-time traffic. Its configuration synthesis requires routing the flows and allocating the bandwidth for Credit Based Shaping (CBS). Current approaches typically employ deadline-oblivious bandwidth allocation and rely on routing to establish deadline guarantees. But the worst-case delay of AVB flows requires complex analysis. Thus, routing integrated with the delay analysis either supports limited Stream Reservation (SR) classes or suffers significant timing overhead. To enable efficient run-time flow setup, we propose SynAVB, a tool to synthesize the configuration of Ethernet AVB, which establishes deadline awareness in bandwidth allocation. SynAVB supports an arbitrary number of SR classes and processes them one by one. For the flows in an SR class, it first performs a deadline-oblivious routing based on Mixed Integer Linear Programming (MILP) to guarantee the necessary bandwidth required by the flows. Then, a deadline-aware slope allocation algorithm computes the bandwidth required on every link to satisfy the deadline of all flows. Our experiment demonstrates that, given the same routes, the deadline-aware bandwidth allocation results in fewer flows violating the deadline compared with other allocation strategies. Moreover, SynAVB can guarantee the deadline of more flows while its run-time is up to 14.3x faster than the state-of-the-art approaches.**

I. Introduction

With the proliferation of automotive applications and sensor traffic, the growing demand for communication bandwidth pushes the traditional field buses to their capacity limit. In response, Audio Video Bridging (AVB) of Ethernet [1] is developed to introduce the Ethernet standard into the automotive domain. Ethernet AVB provides the Quality-of-Service (QoS) guarantee to soft real-time data, in particular multimedia audio and video. Besides the bandwidth that grows with technology, it promises the adoption of internet protocols as the basis of open networks. Thus, the need for expensive and domain-specific technologies can be avoided.

AVB flows in current automotive Ethernet are typically soft real-time, i.e., frames that violate deadlines will be dropped causing a negative impact on the QoS. In Ethernet AVB, the bounded latency is achieved via class-based queuing and Credit Based Shaping (CBS) [1], i.e., CBS limits the transmission of the frames in each class according to the pre-allocated bandwidth specified as the idle slope. Flows with different transit time requirements are assigned to different Stream Reservation (SR) classes. Most of the current AVB networks support up to two SR classes [2], SR A and B. However, the fast-growing sensor traffic with diverse timing requirements may demand more SR classes. According to the IEEE 802.1Qbv standard [3], there can be up to eight queues that perform CBS on an Ethernet port, i.e., up to eight SR classes can be supported. Thus, several works have analyzed Ethernet AVB with arbitrary SR classes [4], [5].

The configuration synthesis of Ethernet AVB involves routing the flows and allocating the bandwidth (which is done by setting a parameter of CBS named idle slope for each SR class) to provide end-to-end delay guarantees. Deadline violations will be counted as heavy penalties in the cost metrics [6]. Meanwhile, as automotive networks evolve towards Software Defined Networking (SDN), fast synthesis algorithms are required for rapid flow setup. Existing solutions are typically based on deadline-oblivious bandwidth allocation. They either statically partition the bandwidth before routing [6]–[8] or allocate the necessary bandwidth when flow routes are given [9]. These techniques have two major limitations. First, bandwidth might be under-allocated to specific SR classes, so the route selection becomes more restricted. Second, the deadline must be considered during routing. In networks with arbitrary SR classes, the worst-case delay is estimated using complex algorithms that evaluate the entire network [9]. A change in one of the flows can impact the delay of all flows, so the number of route candidates whose delay needs to be checked grows exponentially with the number of flows. Although existing solutions [6], [8] employ meta-heuristic algorithms to improve search efficiency, their run-time is still unacceptable for SDN applications.

Our insight is that the deadline can be addressed by the bandwidth allocation instead of routing. As a polynomial-time Deadline-Aware Slope Allocation (DASA) algorithm can be derived from the busy period analysis, it removes the concern of deadline from routing and simplifies it into a Mixed Integer Linear Programming (MILP) problem. Based on this idea, we propose SynAVB, a tool to synthesize routes and idle slopes for Ethernet AVB with arbitrary SR classes. SynAVB processes SR classes in the decreasing order of priority. For each SR class, it first performs deadline-oblivious routing which guarantees the necessary bandwidth for the flows. Then, the DASA algorithm ensures that flows meet their deadline. In scenarios where deadline guarantees cannot be provided due to insufficient remaining bandwidth of the links, it reports the flows that do not have deadline guarantees so that applications

This research was supported through PENTA project HIPER 181004 on high performance vehicle computer (HPVC) and communication system for autonomous driving.

can define their handling procedures.

Our experiments on two industrial topologies demonstrate that, given the flow routes, DASA results in fewer flows that violate deadlines. Compared with existing approaches, SynAVB can be up to 14.3x faster while satisfying the deadlines of more flows. We experimentally compare two routing objectives, Shortest Path (SP) and Load Balancing (LB). The results show that LB routing can set up more flows while SP routing consumes less time.

The paper is structured as follows. Section II discusses the related works. Section III introduces our model of Ethernet AVB and SynAVB. Section IV and V discuss the proposed routing and slope allocation algorithms in detail. Section VI evaluates SynAVB and Section VII concludes.

II. Related Work

Two bandwidth allocation strategies have been widely used in the current Ethernet AVB systems. Some systems, e.g., the ADAS in [10], configure the idle slopes as the bandwidth requested by each SR class. Another common practice is to assume the idle slopes as constants identical for each link and specified by the network designer [6]–[8]. Both strategies lack the awareness of deadlines so bandwidth could be under-allocated to some SR classes. To our knowledge, bandwidth allocation in Ethernet AVB with the awareness of deadlines has not been presented in the literature.

A consequence of the current deadline-oblivious bandwidth allocation is that routing must guarantee the flow deadlines. Deadline-aware routing has been widely studied in real-time or vehicular SDN [11], [12] where the delay on each link can be bounded by weight. However, Ethernet AVB requires different approaches, because its worst-case delay needs specific analysis. As revealed by several analyses of Ethernet AVB [9], [13], [14], the worst-case delay under CBS depends on all of the flows queued for the same link as well as their arrival jitters. Since these facts can be determined only when routes are specified, allocating weights on links to bound the worst-case delay is not feasible. Therefore, deadline-aware routing has been specifically studied for Ethernet AVB. For SR A flows whose delay can be formulated into linear constraints, routing can be solved as a MILP problem [15], [16]. But with arbitrary SR classes, the delay must be achieved by complex analysis, e.g., Forward End-to-End Delay Analysis (FA) [9]. Integrating such complex delay computation into an efficient routing algorithm is difficult. The GRASP algorithm [6], [8] proposes route candidates stochastically and checks for a good solution. Because the search space of routes grows exponentially with the number of flows, the algorithm requires a long execution time, e.g., in the scale of 15 minutes [6]. Thus, it is not suitable for run-time flow setup. Instead, SynAVB ensures flow deadlines through bandwidth allocation. Hence, routing of arbitrary SR classes can be simplified into a MILP problem to be solved efficiently.

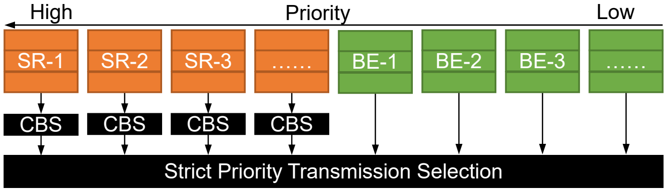


Fig. 1: Traffic shaping of Ethernet AVB

III. System Model

This section discusses our system model and the synthesis flow of SynAVB. We model a network as a directed graph $G = (V, E)$ whose nodes (V) represent switches and hosts. The edges (E) represent network links. A directed edge $(u, v) \in E$ denotes the simplex connection from node u to node v with bandwidth $bw_{u,v}$. The maximum bandwidth that can be allocated to AVB flows is $\gamma bw_{u,v}$. The default value of γ in the AVB standard [1] is 0.75. The delay on a link due to switch processing and signal propagation can be modeled by a constant $pd_{u,v}$ [11]. $in(v)$ is the set of nodes $u \in V$ so that $(u, v) \in E$, and $out(v)$ is the set of nodes $w \in V$ so that $(v, w) \in E$.

A. Credit Based Shaping and Flows

Fig. 1 illustrates the traffic shaping in Ethernet AVB. Standard Ethernet switches can support up to eight traffic classes [3]. $X = \{1, 2, ...\}$ is the set of all SR classes in which smaller numbers refer to SR classes with higher priorities. Each SR class is associated with an egress queue. The remaining queues are allocated to the Best Effort (BE) traffic. CBS controls the transmission of each queue with a credit. The bandwidth allocated to an SR class $x \in X$ on link $(u, v) \in E$ is specified in terms of *idle slope* $\alpha^x_{u,v}$ in bits per second. The corresponding *send slope* is $\beta^x_{u,v} = \alpha^x_{u,v} - bw_{u,v}$. Since the bandwidth used by AVB flows is limited by γ, $\sum_{x \in X} \alpha^x_{u,v} \leqslant \gamma bw_{u,v}$ must hold on all links. The functionality of CBS can be described as follow.

- CBS allows non-empty queues with non-negative credit to transmit. When multiple such queues exist, the one with the highest priority is selected.
- When a queue has frames but does not transmit, its credit increases at the rate of the idle slope.
- When a queue is transmitting, its CBS credit decreases at the rate of the send slope.
- When a queue with positive credit becomes empty, its credit is reset to zero. It is referred to as the *credit reset*.

AVB flows are streams of unicast or multicast frames. F represents the set of all flows in the network. $F^x \subseteq F$ represents the set of class x flows. $F^x_{u,v} \subseteq F$ represents the set of class x flows traversing link (u, v) when flow routes are specified. We target Ethernet AVB with Centralized Network Configuration (CNC) [17]. To request transmission, applications specify a flow f in terms of its source $s_f \in V$, the set of its destinations $D_f \subseteq V$, the maximum frame size c_f (header and overhead included), frame generation period T_f, and its deadline DL_f. We define c^x as the maximum frame

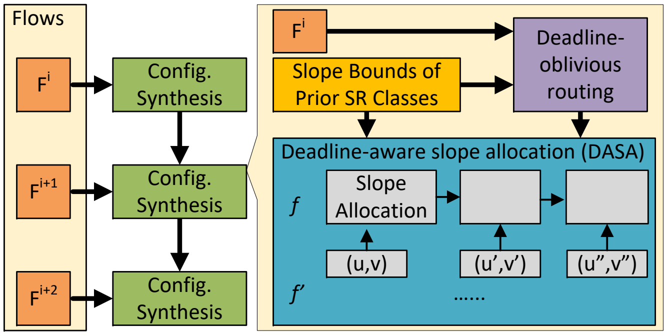


Fig. 2: SynAVB route and slope synthesis flow

size of class x flows and $\overline{c^x}$ as the maximum frame size of all classes with less priority than x, including BE classes. The CNC is responsible for finding the flow route $R_f \subseteq E$ which is an ordered set of links starting from s_f through which frames of f can reach their destinations. It can also adjust the idle slopes to ensure that flows meet their deadlines.

B. An Overview of SynAVB Synthesis Flow

The route and slope synthesis flow proposed in SynAVB can be illustrated in Fig. 2. Given the set of flows F to be set up, it iteratively performs configuration synthesis for each SR class in the decreasing order of priority. For flows in each SR class, it applies deadline-oblivious routing to ensure that all links on the route hold the necessary bandwidth for the flows. After routing, DASA computes the idle slope of every link to guarantee that all flows satisfy their deadline. In scenarios where a link does not have sufficient bandwidth, DASA first allocates the remaining bandwidth to the SR class to minimize the delay on that link. Because other links might have remaining delay budget, these flows might still meet their deadline. Then, after allocating the bandwidth for the entire network, it checks the flows without deadline guarantees and reports them to the CNC. Because AVB flows are typically soft real-time, low criticality applications may still use the configuration without guarantee. Otherwise, if the reduced QoS is unacceptable, CNC rejects these flow requests which informs the applications to perform service degradation [18], e.g., the affected application might be shut down safely.

IV. Deadline-oblivious Routing

Except for the SR class with the highest priority, the end-to-end delay in Ethernet AVB is a complex non-linear function [9]. Thus, existing MILP-based deadline-aware routing algorithms [15], [16] can cover only SR-A flows. Instead, our deadline-oblivious routing applies to an arbitrary number of SR classes and only enforces the necessary condition of schedulability [19], i.e., the idle slope should not be lower than the bandwidth requested by the SR class as stated by Eq. 1. Thus, it needs to be combined with the DASA to provide service guarantees. This section introduces the proposed deadline-oblivious routing in detail.

$$\forall x \in X, (u,v) \in E, \sum_{f \in F^x_{u,v}} \frac{c_f}{T_f} \leqslant \alpha^x_{u,v} \tag{1}$$

As SynAVB adds SR classes to the network incrementally, routing receives the following input in every iteration: the network G, an SR class x, the flows in the SR class F^x, and the idle slopes allocated for all class y with higher priority than x (i.e., $y < x$) on all links $(u,v) \in E$. We propose deadline-oblivious routing with two objectives, LB and SP, which we compare experimentally later.

A. Load Balancing Routing

LB focuses on minimizing the congested links so that the flows in lower priority classes can potentially have more alternative paths. We define the following variables.

- Link occupation $lo^f_{u,v} \in \{0,1\}, \forall (u,v) \in E, f \in F$, is 1 when f is routed on (u,v). Otherwise, it is 0.
- Destination counter $dc^f_{u,v} \in \mathbb{N}, \forall (u,v) \in E, f \in F$ is the number of destinations of f reached over (u,v).
- Bandwidth utilization $bu_{u,v} \in \mathbb{R}$ is the bandwidth on (u,v) utilized by the current flows.
- Maximum utilization $mu \in \mathbb{R}$ is the maximum bandwidth utilization among all of the links in the network.

Note that we define the variables for all $f \in F$. However, routing is performed iteratively for each SR class (F^x).

Objective. *The objective of LB routing is shown in Eq. 2. The primary objective is to balance the network load, i.e., minimize the maximum utilization of the links. So, congested links are minimized. Its secondary objective with weight δ is to minimize the length of the routes to reduce latency.*

$$Min \left\{ mu + \delta \sum_{(u,v) \in E} \sum_{f \in F^x} lo^f_{u,v} \right\} \tag{2}$$

Constraint 1. *The destination counter is zero when a link is not used by a flow. Otherwise, it is lower than the number of destinations of the flow.*

$$\forall f \in F^x, (u,v) \in E: \; dc^f_{u,v} - |D_f| lo^f_{u,v} \leqslant 0 \tag{3}$$

Constraint 2. *Routes must consist of consecutive links. Hence, the sum of the destination counters is $|D|$ for the output links of the source. Each time the route passes a destination, it reduces by one. Otherwise, the sum of the destination counters must be equal for the input and output links of a node.*

$$\begin{aligned}
&\forall f \in F^x, v \in V, \\
&\mathit{if}\; v = s_f: \sum_{\forall u \in in(v)} dc^f_{u,v} = 0, \sum_{\forall u \in out(v)} dc^f_{v,u} = |D_f| \\
&\mathit{if}\; v \in D_f: \sum_{\forall u \in in(v)} dc^f_{u,v} - \sum_{\forall u \in out(v)} dc^f_{v,u} = 1 \\
&\mathit{if}\; v \notin \{s\} \cup D_f: \sum_{\forall u \in in(v)} dc^f_{u,v} - \sum_{\forall u \in out(v)} dc^f_{v,u} = 0
\end{aligned} \tag{4}$$

Constraint 3. *The bandwidth utilized on each link is determined by the flow routes.*

$$\forall (u,v) \in E: \; bu_{u,v} - \sum_{f \in F^x} \frac{c_f lo^f_{u,v}}{bw_{u,v} T_f} = 0 \tag{5}$$

Constraint 4. *Every link must hold the necessary bandwidth according to Eq. 1, i.e., the remaining bandwidth must be larger than the utilized bandwidth.*

$$\forall (u,v) \in E:\ bu_{u,v} \leqslant \gamma bw_{u,v} - \sum_{y=1}^{x-1} \alpha^{y}_{u,v} \tag{6}$$

Constraint 5. *Maximum utilization is the maximum bandwidth that has currently been allocated on all of the links.*

$$\forall (u,v) \in E:\ bu_{u,v} + \sum_{y=1}^{x-1} \alpha^{y}_{u,v} - mu \leqslant 0 \tag{7}$$

B. Shortest Path Routing

The SP routing minimizes the route length which potentially reduces flow latency and network utilization. It can be implemented by removing variables and constraints from the LB routing. Thus, it is simpler and faster. The variables of the SP routing are the link occupation, destination counter, and bandwidth utilization as in LB routing. Its objective is shown in Eq. 8 and its constraints are constraint 1-4.

$$\text{Min}\left\{ \sum_{(u,v)\in E} \sum_{f\in F^{x}} lo^{f}_{u,v} \right\} \tag{8}$$

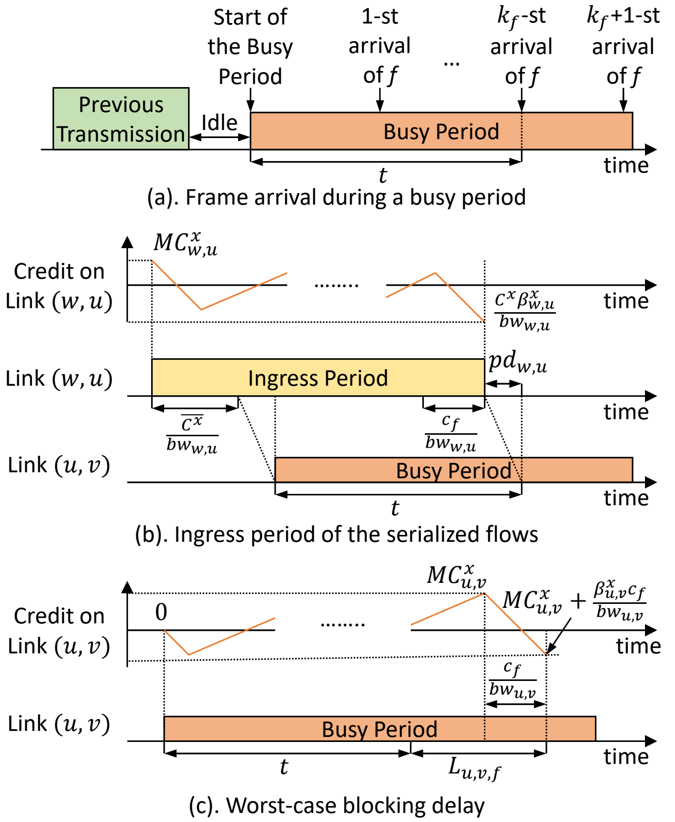
V. Deadline-Aware Slope Allocation

DASA partitions the end-to-end deadlines over routes of the flows resulting in per-link deadlines. Then, idle slopes are computed for each link so that flows meet their per-link deadlines. To compute such slopes, we perform busy period analysis to find the correlation between idle slopes and worst-case delay. Our busy period analysis is based on three theorems. Theorem 1 and 2 are two upper bounds of the blocking that a flow might experience. Theorem 3 revels their connection with the worst-case delay. We solve the equations in theorems which leads to the idle slope calculation in DASA. In this section, we first introduce the background of delay analysis in Ethernet AVB then discuss our busy period analysis in detail. The slope allocation algorithm is presented at last.

A. Background of the Delay Analysis in Ethernet AVB

When frames of the AVB flows are queued for a link, they experience a series of blocking over their routes. On each node, we define the *maximum and minimum traversal time*, represented by $Dmax_{v,f}$ and $Dmin_{v,f}$, as the maximum and minimum time frames of f can reach v from s_f. They can be iteratively calculated with Eq. 9 [9], in which $L_{u,v,f}$ is the maximum delay of f's frames caused by queuing and transmission on link (u,v). $Dmin_{s_f,f}$ and $Dmax_{s_f,f}$ are defined to be zero. The *jitter* is *the difference between the maximum and minimum traversal time*, i.e., $J_{v,f} = Dmax_{v,f} - Dmin_{v,f}$.

$$\begin{aligned} Dmin_{v,f} &= Dmin_{u,f} + \frac{c_f}{bw_{u,v}} + pd_{u,v} \\ Dmax_{v,f} &= Dmax_{u,f} + L_{u,v,f} + pd_{u,v} \end{aligned} \tag{9}$$



(a). Frame arrival during a busy period

(b). Ingress period of the serialized flows

(c). Worst-case blocking delay

Fig. 3: Busy period under CBS

The busy period is the duration in which the queue of SR class x is non-empty. The goal of our busy period analysis is to *allocate an idle slope* $\alpha^x_{u,v}$ *so that* $L_{u,v,f} + pd_{u,v}$ *is bounded by a deadline* $DL_{u,v,f}$. Since the blocking on a link causes arrival jitter on subsequent links, links are analyzed in a topological order to consider the arrival jitter, i.e., links with higher ranks in the routes must be analyzed in prior [10]. When link (u,v) is reached, the following parameters are known.

- The flows in SR class x routed on link (u,v) $(F^x_{u,v})$
- The idle slopes for SR classes with higher priority than x on all links of the network
- The idle slopes $\alpha^x_{w,u}, \forall w \in in(u)$.
- The arrival jitter $J_{u,g}, \forall g \in F^x_{u,v}$.

The maximum CBS credit during the busy period, denoted as $MC^x_{u,v}$, has been studied in [5]. It can be calculated by Eq. 10 [5], which we use in our analysis.

$$MC^x_{u,v} = \frac{\alpha^x_{u,v}(bw_{u,v}\overline{c^x_{u,v}} - \sum_{y=1}^{x-1}\beta^y_{u,v}c^y_{u,v})}{bw_{u,v}(bw_{u,v} - \sum_{y=1}^{x-1}\alpha^y_{u,v})} \tag{10}$$

B. Busy Period Analysis under CBS

Our busy period analysis allocates an idle slope $\alpha^x_{u,v}$ so that $L_{u,v,f} + pd_{u,v}$ is bounded by a deadline $DL_{u,v,f}$. Fig. 3(a) illustrates a busy period. It starts after an idle period in which the CBS credit is (reset to) zero. For the analysis, we define the following variables. Without loss of generality, we assume that the k_f-st frame of f suffers the worst-case delay.

- t is the duration between the arrival of the k_f-st frame of f at node u and the start of the busy period.
- $\forall g \in F^x_{u,v} \setminus \{f\}$, k_g is the number of the frames that arrive during $[0, t]$ of flow g.

The *workload* of a flow $g \in F$ is $k_g c_f$. The workloads of the flows are limited by the CBS on upstream links. This is referred to as the serialization effect [9]. Existing works explore the serialization effect to improve the tightness of the delay estimation. Instead, we use it to bound the idle slopes. We define the serialized flows set $S^w_{u,v}$ for all $w \in in(u)$ to be $\{f \in F_{u,v} | (u, v) \in R_f \wedge (w, u) \in R_f\}$. Flows sourced from node u does not belong to any serialized flows set. The set of these flows is denoted by $Q_{u,v}$.

Theorem 1. *The workloads of the flows are bounded by their jitters and periods as shown in Eq. 11 ($\forall w \in in(u)$) and 12.*

$$\sum_{g \in S^w_{u,v}} k_g c_g \leqslant \sum_{g \in S^w_{u,v}} \frac{c_g}{T_g} t + \sum_{g \in S^w_{u,v}} c_g \Big(1 + \frac{J_{v,g}}{T_g}\Big) \tag{11}$$

$$\sum_{g \in Q_{u,v}} k_g c_g \leqslant \sum_{g \in Q_{u,v}} \frac{c_g}{T_g} t + \sum_{g \in Q_{u,v}} c_g \Big(1 + \frac{J_{v,g}}{T_g}\Big) \tag{12}$$

Proof. The number of frames that arrive in $[0, t]$ is limited by the request bound function in Eq. 13 [9]. □

$$\forall g \in F_{u,v},\ k_g \leqslant 1 + \frac{t + J_{v,g}}{T_g} \tag{13}$$

Theorem 2. *The workloads of the serialized flows that arrive during $[0, t]$ are limited by the CBS as shown in Eq. 14. Note that $MC^x_{w,u}$ can be calculated by Eq. 10.*

$$\begin{aligned}
&\forall w \in in(u), \textit{if } f \in S^w_{u,v} : \\
&\sum_{g \in S^w_{u,v}} k_g c_g \leqslant \alpha^x_{w,u} t + MC^x_{w,u} + \frac{\alpha^x_{w,u} c^x - \beta^x_{w,u} c_f}{bw_{w,u}} \\
&\textit{else :} \\
&\sum_{g \in S^w_{u,v}} k_g c_g \leqslant \alpha^x_{w,u} t + MC^x_{w,u} + c^x
\end{aligned} \tag{14}$$

Proof. As illustrated by Fig. 3(b), the duration in which the frames transmitted over (w, u) can reach node u within the duration $[0, t]$ is referred to as the ingress period. Its maximum length is $t + \frac{c^x}{bw_{w,u}}$. When $f \notin S^w_{u,v}$, the worst-case ingress period starts with the maximum credit $MC^x_{w,u}$ and ends with the minimum credit (which occurs after the transmission of a maximum-sized frame). Hence, the credit replenished during the ingress period has a lower bound (Eq. 15). Note that credit reset and the transmission of the flows in $F_{w,u} \setminus F_{u,v}$ will only increase the Left-Hand Side (LHS) of the inequality.

$$\begin{aligned}
&\alpha^x_{w,u}\Big(t + \frac{c^x}{bw_{w,u}} - \sum_{g \in S^w_{u,v}} \frac{k_g c_g}{bw_{w,u}}\Big) + \beta^x_{w,u} \sum_{g \in S^w_{u,v}} \frac{k_g c_g}{bw_{w,u}} \\
&\geqslant \frac{c^x \beta^x_{w,u}}{bw_{w,u}} - MC^x_{w,u}
\end{aligned} \tag{15}$$

When $f \in S^w_{u,v}$, the ingress period always ends with the transmission of the k_f-st frame of f. So, the minimum credit at the end of the ingress period is $\frac{\beta^x_{w,u} c_f}{bw_{w,u}}$. Simplifying the equations with $\alpha^x_{w,u} - \beta^x_{w,u} = bw_{w,u}$ leads to Eq. 14. □

Theorem 3. *The $L_{u,v,f}$ is upper bounded by Eq. 16.*

$$\begin{aligned}
L_{u,v,f} \leqslant &\sum_{g \in F_{u,v}} \frac{k_g c_g}{\alpha^x_{u,v}} - t + \frac{c_f \beta^x_{u,v}}{\alpha^x_{u,v} bw_{u,v}} + \\
&\frac{(\overline{c^x_{u,v}} + \sum_{y=1}^{x-1} c^y_{u,v})}{bw_{u,v} - \sum_{y=1}^{x-1} \alpha^y_{u,v}}
\end{aligned} \tag{16}$$

Proof. The upper bound of the ratio between $MC^x_{u,v}$ and $\alpha^x_{u,v}$ can be derived from Eq. 10 by relaxing $\beta^y_{u,v}$ to $-bw_{u,v}$.

$$\frac{MC^x_{u,v}}{\alpha^x_{u,v}} \leqslant \frac{\overline{c^x_{u,v}} + \sum_{y=1}^{x-1} c^y_{u,v}}{bw_{u,v} - \sum_{y=1}^{x-1} \alpha^y_{u,v}} \tag{17}$$

In the worst-case scenarios, the k_f-st frame of f initiates transmission when the queue experiences maximum blocking from other SR classes, i.e., it holds maximum credit. This is illustrated in Fig. 3(c). Since there is no credit reset in the duration $t + L_{u,v,f}$, the credit replenishment is upper bounded by Eq. 18. Using Eq. 17 to simplify Eq. 18 leads to Eq. 16. □

$$\begin{aligned}
&\alpha^x_{u,v}\Big(t + L_{u,v,f} - \sum_{g \in F_{u,v}} \frac{k_g c_g}{bw_{u,v}}\Big) + \beta^x_{w,u} \sum_{g \in S^w_{u,v}} \frac{k_g c_g}{bw_{w,u}} \\
&\leqslant MC^x_{u,v} + \frac{c^f \beta^x_{u,v}}{bw_{u,v}}
\end{aligned} \tag{18}$$

Solving the equations: for conciseness, we define the abbreviations in Eq. 19 for the coefficient and offset of t in the Right-Hand Side (RHS) of Eq. 11, 12, and 14, which are constants known from the input.

$$\begin{cases}
\text{RHS of Eq. 11} = Aj_w t + Bj_w, \forall w \in in(u) \\
\text{RHS of Eq. 12} = Aq t + Bq \\
\text{RHS of Eq. 14} = As_w t + Bs_w, \forall w \in in(u)
\end{cases} \tag{19}$$

To solve for $\alpha^x_{u,v}$, we bound a function of t and $\alpha^x_{u,v}$ with $D_{u,v,f}$. According to Theorem 1 and 2, the RHS of Eq. 20 $\geqslant \sum_{g \in F_{u,v}} \frac{k_g c_g}{\alpha^x_{u,v}} - t$. Hence, if Eq. 20 holds for all $t > 0$, $D_{u,v,f} - pd_{u,v} \geqslant L_{u,v,f}$ according to Theorem 3.

$$\begin{aligned}
&D_{u,v,f} - \frac{c_f \beta^x_{u,v}}{\alpha^x_{u,v} bw_{u,v}} - \frac{(\overline{c^x_{u,v}} + \sum_{y=1}^{x-1} c^y_{u,v})}{bw_{u,v} - \sum_{y=1}^{x-1} \alpha^y_{u,v}} - pd_{u,v} \\
&\geqslant \Bigg(\frac{Aq}{\alpha^x_{u,v}} - \frac{\sum_{g \in Q_{u,v}} c_g/T_g}{\sum_{g \in F_{u,v}} c_g/T_g}\Bigg) t + \frac{Bq}{\alpha^x_{u,v}} + \\
&\sum_{w \in in(u)} \min \Bigg\{ \underbrace{\Bigg(\frac{Aj_w}{\alpha^x_{u,v}} - \frac{\sum_{g \in S^w_{u,v}} c_g/T_g}{\sum_{g \in F_{u,v}} c_g/T_g}\Bigg) t + \frac{Bj_w}{\alpha^x_{u,v}}}_{\text{J-term}}, \\
&\underbrace{\Bigg(\frac{As_w}{\alpha^x_{u,v}} - \frac{\sum_{g \in S^w_{u,v}} c_g/T_g}{\sum_{g \in F_{u,v}} c_g/T_g}\Bigg) t + \frac{Bs_w}{\alpha^x_{u,v}}}_{\text{S-term}} \Bigg\}
\end{aligned} \tag{20}$$

With a given $\alpha^x_{u,v}$, the LHS of Eq. 20 is a constant while the RHS is a function of t. The inequality holds if the maximum of the function does not exceed the constant. Thus, we study the monotonicity of the function to calculate $\alpha^x_{u,v}$. Since Eq. 1 is enforced by routing, the coefficient of J-term is non-positive, i.e., it does not increase with t. The coefficient of S-term may vary between links. If $\frac{Bs_w - Bj_w}{Aj_w - As_w} \geqslant 0$ for a w, the J-term equals the S-term when $t = \frac{Bs_w - Bj_w}{Aj_w - As_w}$ and their minimum is a non-increasing function for $t > \frac{Bs_w - Bj_w}{Aj_w - As_w}$. We define $TI_u = [t_0, t_1 ..., t_{|TI_u|-1}]$ as *the increasingly ordered sequence of positive* $\frac{Bs_w - Bj_w}{Aj_w - As_w}$ *for all* $w \in in(u)$. The monotonicity of the RHS of Eq. 20 with t can be described as follow.

- The function is non-increasing on $[t_{|TI_u|-1}, +\infty)$.
- For intervals $[0, t_0]$, $[t_0, t_1]$, ..., $[t_{|TI_u|-2}, t_{|TI_u|-1}]$, the function is linear. Thus, the maximum can be achieved on the boundaries of the intervals.

Therefore, an idle slope can be found by checking all possible maximum of the function so that Eq. 20 holds, i.e., it can be calculated by Eq. 21.

$$\alpha^x_{u,v} = \max_{t\in\{0\}\cup TI_u} \Big\{ \Big(-c_f + Aqt + Bq + \sum_{w\in in(u)} \min\Big\{Aj_w t + Bj_w, As_w t + Bs_w\Big\}\Big) \Big/ \Big(D_{u,v,f} + t - \frac{c_f}{bw_{u,v}} - \frac{(\overline{c^x_{u,v}} + \sum_{y=1}^{x-1} c^y_{u,v})}{bw_{u,v} - \sum_{y=1}^{x-1}\alpha^y_{u,v}} - pd_{u,v}\Big)\Big\} \tag{21}$$

On the other hand, when the idle slope is given, the $L_{u,v,f}$ can be calculated by Eq. 22.

$$L_{u,v,f} = \frac{1}{\alpha^x_{u,v}} \max_{t\in\{0\}\cup TI_u} \Big\{Aqt + Bq - t + \sum_{w\in in(u)} \min\Big\{Aj_w t + Bj_w, As_w t + Bs_w\Big\}\Big\} + \frac{c_f \beta^x_{u,v}}{\alpha^x_{u,v} bw_{u,v}} + \frac{(\overline{c^x_{u,v}} + \sum_{y=1}^{x-1} c^y_{u,v})}{bw_{u,v} - \sum_{y=1}^{x-1}\alpha^y_{u,v}} \tag{22}$$

C. Idle Slope Allocation

As a common practice (e.g., in the AVB profile [2]), the end-to-end deadline can be uniformly distributed over the entire route resulting in equal per-link deadlines. Hence, DASA (Algorithm 1) allocates idle slopes based on the strategy in Eq. 21. It initializes the idle slopes to zero and equally partition the end-to-end deadline on the routed links (line 1-2). Algorithm 1 processes links in the same topological order as in FA; the links are sorted based on their maximum ranks in the flow routes (*Sorted_Links*) [10]. On every edge, the algorithm allocates an idle slope so that all flows in $F^x_{u,v}$ meet their per-link deadline. This can be achieved by computing an idle slope for every flow using Eq. 21 and taking their maximum. For each flow, the algorithm first computes TI_u according to its definition using *S_J_Intersect* (line 5). Then, *Slope_Alloc* computes Eq. 21 (line 6). Once an idle slope is specified for a link, *Next_Hop_Jitter* computes the worst-case delay of all flows on that link via Eq. 22 and updates the arrival jitter of the next hop via Eq. 9 (line 12). Thus, when the analysis propagates through the entire network, the worst-case end-to-end delay for all of the flows in F^x is known.

Algorithm 1: The DASA algorithm

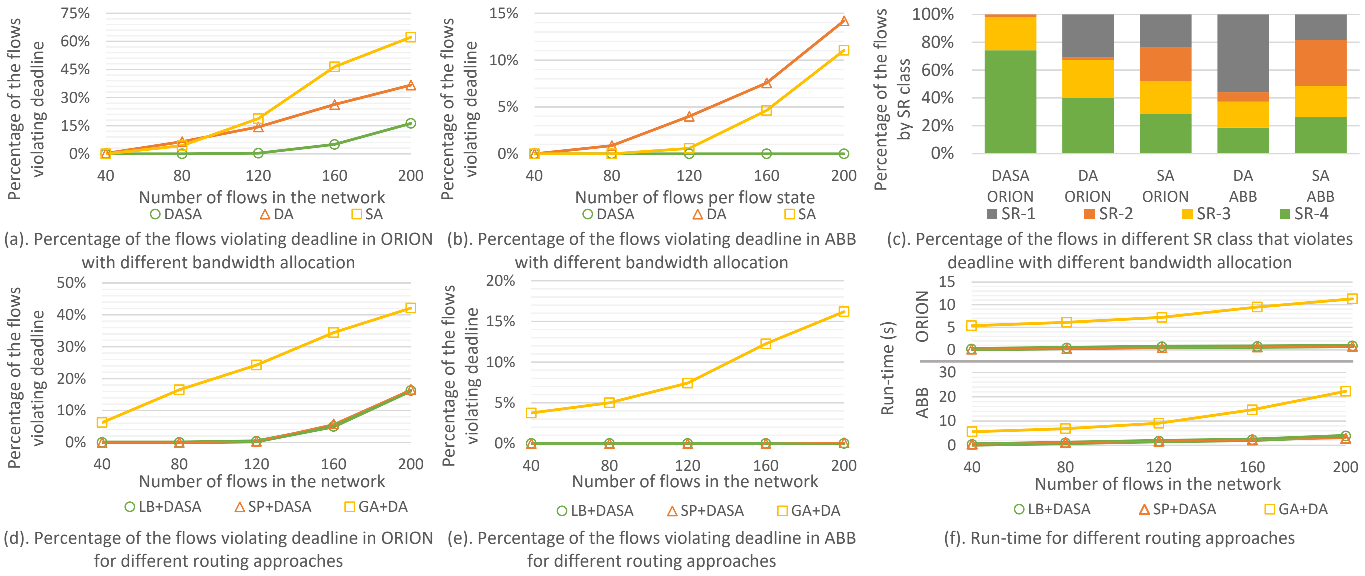
```
   input : The network $G$; Routed flows $F^x$; Idle slopes
           for high-priority SR classes on all links
   output: Idle slopes for class $x$ on all links ($\alpha^x_{u,v}$);
           Flows without deadline guarantee ($Fv$)
 1 $\forall(u,v) \in E, \alpha^x_{u,v} = 0$
 2 $\forall(u,v) \in E, \forall f \in F^x, DL_{u,v,f} = DL_f / \max\{|P_{d,f}|\}$
 3 for $(u,v) \in Sorted\_Links(G, F^x)$ do
 4     for $f \in F^x_{u,v}$ do
 5         $TI_u = S\_J\_Intersect(f, F^x_{u,v})$
 6         $\alpha = Slope\_Alloc(f, F^x_{u,v}, TI_u)$
 7         if $\alpha + \sum_{y=1}^{x-1} \alpha^y_{u,v} \geqslant \gamma bw_{u,v}$ then
 8             $\alpha = \gamma bw_{u,v} - \sum_{y=1}^{x-1} \alpha^y_{u,v}$
 9         end
10         $\alpha^x_{u,v} = max(\alpha^x_{u,v}, \alpha)$
11     end
12     $Next\_Hop\_Jitter(F_{u,v}, \alpha^x_{u,v})$
13 end
14 $Fv = \emptyset$
15 for $f \in F^x$ do
16     if $\exists d \in D_f, Dmax_{d,f} \geqslant DL_f$ then
17         $Fv = Fv \cup \{f\}$
18     end
19 end
```

When a link does not have the sufficient bandwidth left by the high-priority classes, the algorithm allocates all the available bandwidth of the link (line 7-9) based on two major reasons. First, this minimizes the delay of the flows on that link. Since the time budget might not be fully consumed on other links, it is still possible for these flows to meet their deadline. Second, if deadline violation occurs and applications can tolerate the reduced QoS, allocating more bandwidth leads to lower average latency and fewer frame losses. When the algorithm finishes, the flows without service guarantees can be reported as Fv (line 14-19). It is up to the applications to determine whether they will tolerate the reduction of QoS or initiate service degradation.

The complexity for calculating the parameters in Eq. 19 is $|F|$ and the complexity to evaluate all $S^w_{u,v}$ is $|V|$. Thus, the function *S_J_Intersect* has complexity $|F| + |V|$. As all of the flows in $F^x_{u,v}$ needs to be iterated for each $(u,v) \in E$, the algorithm has polynomial run-time $|E||F|(|F| + |V|)$.

VI. Evaluation

We evaluate the proposed route and slope synthesis flow using two topologies: the avionic network in ORION crew exploration vehicle [20] and an Ethernet AVB built for industrial automation by ABB [6]. Both networks are formed with uniform links ($bw_{u,v}$=100Mbps, $pd_{u,v}$=5.21µs). γ is set to 0.75 as the default value in the AVB standard [1]. Since the original applications only contain SR-A flows, we construct test setups using flows with random source and



(a). Percentage of the flows violating deadline in ORION with different bandwidth allocation

(b). Percentage of the flows violating deadline in ABB with different bandwidth allocation

(c). Percentage of the flows in different SR class that violates deadline with different bandwidth allocation

(d). Percentage of the flows violating deadline in ORION for different routing approaches

(e). Percentage of the flows violating deadline in ABB for different routing approaches

(f). Run-time for different routing approaches

Fig. 4: Evaluation result

destination. The specification for each SR class follows the template in Table I adopted from [2]. We select the weight of the secondary objective δ in LB routing based on iterative experiments. When δ is larger than 0.01, the result of LB routing is similar to SP routing. Otherwise, varying it above zero does not cause significant impact. Thus, δ is set to 0.01.

TABLE I: The flow template adopted from [2]

SR Class	T_f (µs)	$c_f/bw_{u,v}$ (µs)	DL_f (ms)
SR-1	125	9.28	2
SR-2	250	11.2	10
SR-3	1333.33	87.2	15
SR-4	1451.25	87.2	15

A. Evaluation of the Slope Allocation

To evaluate our DASA algorithm, we compare its performance with the dynamic allocation (DA) [9] in which the allocated bandwidth on every link equals the requested bandwidth of each SR class. We also compare with the static allocation (SA). In [6], all of the available bandwidth is allocated to the only SR class in the network. We generalize it by partitioning the bandwidth based on the ratio of the requested bandwidth of each SR class. For instance, when there is one flow in each SR class, the ratio is $\frac{9.28}{125} : \frac{11.2}{250} : \frac{87.2}{1333.33} : \frac{87.2}{1451.25}$. Thus, the idle slope of SR-1 is $0.30\times\gamma\times bw_{u,v} = 0.23\times bw_{u,v}$.

We generate test cases with 40, 80, 120, 160, and 200 flows. All SR classes have the same number of flows, e.g., there are 50 flows in every SR class for test cases with 200 flows in total. For each number of flows, 10 test cases are generated resulting in 5×10 test cases on each topology. As the metric for performance comparison, we report *the percentage of the flows without deadline guarantees*.

The flow routes are specified by the LB routing. The average percentage of the flows violating the deadline versus the total number of flows in the networks are shown in Fig. 4(a) and 4(b). Because the networks have limited capacity, more deadline violations occur when the number of flows increases. In all of the test cases, DASA always leads to minimum deadline violation due to its awareness of the deadline. For instance, none of the flows violates the deadline in ABB with DASA. In ORION with 160 flows, 5% of the flows violate deadlines with DASA while 26.3% (46.4%) of the flows violate deadlines with DA (SA). When the network load is relatively low, e.g., in ORION with 80 flows and ABB with 120-200 flows, SA leads to slightly less deadline violation than DA. However, when the network load increases, e.g., in ORION with 200 flows, SA causes significantly more flows violating deadlines.

Besides satisfying the deadline of more flows, another advantage of DASA is that bandwidth is first allocated to SR classes with higher priority. As a result, deadline violation mostly occurs in SR classes with lower priority. Fig. 4(c) shows the percentage of each SR class for the flows that violate the deadline in both ORION and ABB with 200 flows. Note that no deadline violation occurs in ABB with DASA so it is not plotted. With DASA, no flow from SR-1 violates the deadline and only 1.8% of the flows violating the deadline are from SR-2. Instead, with DA, deadline violation occurs more frequently for SR-1 flows whose deadline is tight, e.g., 30.7% of the flows in ORION and 56% of the flows in ABB are from SR-1. Compared with DA, SA leads to a relatively equal chance of deadline violation for each SR class, e.g., the percentage in ORION is 23.9, 24.2, 23.6, 28.3 for SR 1 to 4. Thus, we conclude that the virtue of DASA is that it guarantees the deadline for more flows and protects the high priority flows when the networks are saturated.

B. Evaluation of the Combined Approaches

We compare the approaches combining deadline-oblivious routing and DASA with the existing deadline-aware routing. We evaluate three approaches. DASA is combined with both

SP and LB routing (SP+DASA, LB+DASA). Also, we compare with the GRASP routing algorithm [6] based on DA (GA+DA). GRASP has two stages: the initialization stage which must be complete to produce a valid solution and the hill-climbing stage where a timeout can be set. The original work set a 15-minute timeout for the entire algorithm [6], which is not feasible for run-time flow setup. Our approaches only consume a few seconds. For a fair comparison, we run the initialization stage and set a 5-second timeout for the hill-climbing stage in GRASP.

Same as the previous evaluation, we generate 50 test cases with 40-200 flows, each containing the same number of flows for every SR class. *A flow is set up successfully if the route and the idle slope found guarantee its deadline.* We run the three approaches to evaluate *the percentage of the flows that are successfully set up as well as their run-time*.

Fig 4(d) and 4(e) show the average percentage of the flows violating the deadline versus the total number of flows for different approaches. Similar to the previous experiments, ABB, which is a larger network, can host more flows without deadline violation. Considering deadline during the slope allocation improves the capability of the routes to provide deadline guarantees. Thus, LB+DASA leads to the fewest deadline violation. That of SP+DASA is similar but slightly higher, e.g, in ORION with 160 flows, the percentage of the flows violating the deadline is 5% for LB+DASA and 5.8% for SP+DASA. It is because routing flows on shortest paths may cause high utilization in some of the links so that there might not be sufficient bandwidth for low priority flows. GA+DA leads to significantly more deadline violations, e.g, in ORION with 160 flows, the flows violating the deadline is 34.4% for GA+DA. The reason is that the GRASP algorithm [8] aims at finding the near-optimal solution by stochastically searching a considerable amount of route candidates. Thus, it has limited usage in scenarios where a low run-time is required.

Fig 4(f) shows the average run-time for different solutions. Compared with GA+DA, deadline-oblivious routing spares the effort of checking a large number of route candidates. So, they are faster and more scalable. SP+DASA is faster than LB+DASA since its MILP model consists of fewer variables and constraints. For instance, in ABB with 200 flows, SP+DASA consumes 2.8s while LB+DASA consumes 3.8s. GA+DA is the slowest among all solutions, while resulting in most deadline violations. For instance, in ORION with 200 flows, LB+DASA takes 0.86s, SP+DASA takes 0.79s, and GA+DA takes 11.3s, i.e., the proposed solution can be up to 14.3x faster. Due to the overwhelming scale of the routing search space, we believe pruning the search space is necessary for approaches targeting online usages.

VII. Conclusion

In this paper, we propose SynAVB to synthesize route and CBS idle slopes for Ethernet AVB with an arbitrary number of SR classes. We establish the deadline-awareness in slope allocation instead of flow routing, i.e., SynAVB performs deadline-oblivious routing and then DASA. We demonstrate that, compared with other common slope allocation approaches, DASA ensures more flows meet their deadline given the same flow routes. Also, it can enforce the deadline of the high-priority flows first when the network is saturated. Compared with the current synthesis approaches, our proposed solution leads to fewer flows that violate deadline while consuming quite shorter run-time. Therefore, it can be efficiently applied for flow setups when the network is on the run. In the future, we are interested in the deadline-awareness in slope allocation with the existence of time-triggered traffic.

References

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

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

[3] "IEEE standard for local and metropolitan area networks-enhancements for scheduled traffic," *IEEE Std 802.1Qbv-2015*, pp. 1–57, 2016.

[4] H. Daigmorte, M. Boyer, and L. Zhao, "Modelling in network calculus a TSN architecture mixing time-triggered, credit based shaper and best-effort queues," Jun. 2018.

[5] E. Mohammadpour, E. Stai, and J.-Y. Le Boudec, "Improved credit bounds for the credit-based shaper in time-sensitive networking," *IEEE Networking Letters*, vol. 1, no. 3, pp. 136–139, 2019.

[6] 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.

[7] G. Alderisi, A. Caltabiano, G. Vasta, G. Iannizzotto, T. Steinbach, and L. L. Bello, "Simulative assessments of IEEE 802.1 Ethernet AVB and time-triggered Ethernet for advanced driver assistance systems and in-car infotainment," in *2012 IEEE VNC*, 2012, pp. 187–194.

[8] 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.

[9] 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.

[10] G. Kemayo, F. Ridouard, H. Bauer, and P. Richard, "A forward end-to-end delays analysis for packet switched networks," in *RTNS2014*, 2014, p. 65–74.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] "IEEE standard for local and metropolitan area networks-stream reservation protocol (SRP) enhancements and performance improvements," *IEEE Std 802.1Qcc-2018*, pp. 1–208, 2018.

[18] T. Ishigooka, S. Otsuka, K. Serizawa, R. Tsuchiya, and F. Narisawa, "Graceful degradation design process for autonomous driving system," *Computer Safety, Reliability, and Security*, pp. 19–34, 2019.

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

[20] 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.