Deadline-Based Class Assignment for Time-Sensitive Network Frame Preemption

Omolade Ikumapayi*, Paul Agbaje†, Yanyan Zhuang*, Habeeb Olufowobi†, Gedare Bloom*
*University of Colorado Colorado Springs, Colorado Springs, CO, USA
†University of Texas at Arlington, Arlington, TX, USA

***Abstract*—The IEEE 802.1Q working group defines the Ethernet time-sensitive networking (TSN) standard to support data-intensive industrial real-time networks. Unfortunately, TSN has the possibility for frame priority inversion that can lead to deadline misses. In this paper, we present a novel approach to address priority inversion in TSN that prioritizes frames during network configuration, determines traffic paths off-line with integer linear programming (ILP), and schedules transmissions on-line using the earliest deadline first (EDF) algorithm. Our approach, the ILP deadline-based TSN (ILP-DTSN), optimizes the network for time-sensitive traffic while minimizing the blocking effects of preemption. ILP-DTSN results in fewer missed deadlines compared with the time-aware shaper (TAS) with one-level preemption while reducing average end-to-end latency by up to 32%.**

***Index Terms*—Time-sensitive networking, Ethernet**

I. Introduction

To support data-rich industrial applications, fieldbus protocols are extended with higher-bandwidth networks including variants of Ethernet. Unfortunately, Ethernet's design does not support global time synchronization or guarantees for message deadlines [1], which are required for real-time systems that deliver time-critical messages with varying quality of service (QoS) requirements. To fulfill the QoS requirements for time-critical traffic, the IEEE 802.1Q working group developed the Ethernet time-sensitive networking (TSN) standard [2] with time synchronization and traffic shaping. The IEEE 802.1Q standard proposes a classification scheme for network traffic, enabling the differentiation of priority levels for TSN to improve the QoS of real-time traffic [3]. According to the IEEE 802.3Qbu standard [4], when frame preemption is supported on a port, the Media Access Control (MAC) provides two MAC service interfaces: a preemptable (pMAC) service interface and an express (eMAC) service interface. The transmission of frames follows two rules: (i) the express frames can preempt the preemptable frames; (ii) frames in the same class cannot preempt each other. Therefore, express frames are never subject to preemption. A MAC merge sublayer between the MAC and Physical layers is connected to the eMAC and pMAC. Each egress port has a maximum of eight queues to allocate the two classes of frames, and each queue is mapped to either the eMAC or the pMAC interface as shown in Figure 1. Time-critical traffic is queued in the eMAC, while frames without critical timing constraints are queued to the pMAC interface.

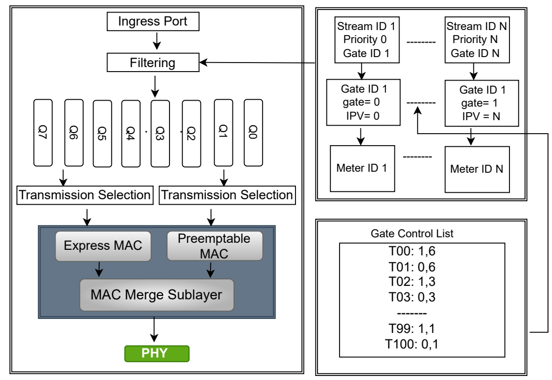


Fig. 1: Switch Queue Model with Frame Preemption.

TSN provides high bandwidth and low latency but can cause priority inversion for medium-priority frames. Priority inversion (blocking) occurs when a low-priority frame holds a resource that a higher-priority frame requires, delaying the higher-priority frame until the low-priority frame releases it. Priority inversion can cause deadline misses for safety-critical messages [5], [6] especially when medium-priority frames have tight deadlines or get blocked at multiple switches.

In this paper, we address priority inversion in TSN with a novel approach that delays low-priority frames that might block medium-priority frames with tight deadlines. Our approach assigns priorities to frames during network configuration and determines network paths using integer linear programming (ILP). The transmission schedule is determined online using the earliest deadline first (EDF) algorithm. This approach reduces the risk of deadline misses.

In this work we make the following contributions:

1) We introduce a novel approach that assigns priorities to frames considering frame preemption and that controls their insertion times into transmission queues.
2) We propose a methodology using ILP for deadline-based TSN (ILP-DTSN) for finding the optimal allocation of frames to the two MAC service interfaces: eMAC and pMAC. ILP-DTSN provides QoS for all priority levels despite TSN frame preemption by using ILP optimization.
3) We evaluate the performance of ILP-DTSN in comparison with Time-Aware Shapers (TAS) and multi-level preemption. ILP-DTSN has fewer missed deadlines and reduces the average end-to-end delay by up to 32% compared to TAS with one-level preemption and up to 9% compared to multi-level preemption.

II. Background and Motivation

This section outlines the time-aware shaper (TAS) and deadline-based priority assignment, and provides a motivating example for ILP-DTSN. Table I summarizes our notation.

A. Time-Aware Shaper (TAS)

The TAS, specified in IEEE 802.1Qbv, enhances QoS in TSN. TAS establishes a Time-Division Multiplexed (TDM) channel for Ethernet, combining multiple network traffic streams by assigning distinct time slots to each stream. Frames are transmitted once all their signal segments have been sent. TAS operates at the egress port of an interface, utilizing time-aware gates controlled by gate control lists (GCL). These gates open and close based on a predefined schedule window, selecting frames from open queues using priority-based or credit-based shapers.

To differentiate traffic, TAS assigns priority levels using the Priority Code Point (PCP) field in the VLAN ID tag of 802.1Q frames. Commonly, traffic is classified into Class A and Class B, where Class A holds a higher priority level (e.g., 3) and Class B has a lower priority level (e.g., 2). Frames from Class A and Class B are chosen based on the open gates, while frames from closed gates are disregarded. TAS effectively minimizes propagation delay and ensures a specific time window for TSN Stream Traffic (ST), which encompasses time-sensitive data such as audio, video, and control messages and necessitates deterministic and bounded end-to-end latency.

B. Deadline-based TSN (D-TSN)

The Per-Stream Filtering and Policing (PSFP) model is a hierarchical standard introduced in IEEE 802.1Qci used to determine queue allocation and ensure reliability in TSN. This model is employed in the deadline-based priority assignments to allow filtering and policing decisions to be made per stream basis as proposed by Patti et al. [7], which we refer to as D-TSN. The PSFP model consists of three hierarchy levels: stream filters for processing frames, stream gates for allowing frames through, and flow meters for flow information.

The stream gate table has attributes such as gate identifier (ID) and internal priority value (IPV), which are used in the mapping of frames to streams in the stream filters. This mapping is based on the PCP calculated during frame generation. The PCP, determined by the absolute deadline, is used by end nodes to insert frames into the source transmission queue [8]. Frames are sorted into n such queues denoted $Q_i \in \{Q_0, Q_1, \ldots, Q_{n-1}\}$, where Q_{n-1} has the highest priority. A TSN switch uses the PSFP stream GCL to determine which traffic queue is authorized to transmit at a particular time, allowing a frame's priority to increase by modifying the queue into which the frame is inserted hop-by-hop. The priority of the queue into which the frame will be inserted is determined by the IPV at time unit u. D-TSN schedules a frame to transmit if it satisfies the following conditions [7]:

$$\begin{cases} d_{i,j} - t > u \\ d_{i,j} - t \leq T_c \end{cases} \tag{1}$$

TABLE I: Notation.

Symbol	Definition
T_c	Total Cycle time
n	Number of queues in each switch
α	Classification for eMAC frames
β	Classification for pMAC frames
C_β	Preemption cost
F_i	The *i-th* flow
$f_{i,j}$	Frame j of flow F_i
C_i	Transmission time of each frame in F_i
T_i	Period of each frame in F_i
D_i	Relative deadline of all frames in F_i
$d_{i,j}$	Absolute deadline of $f_{i,j}$
$a_{i,j}$	Arrival time of $f_{i,j}$
$\rho_{i,j}$	Priority (queue) of $f_{i,j}$ at an egress queue

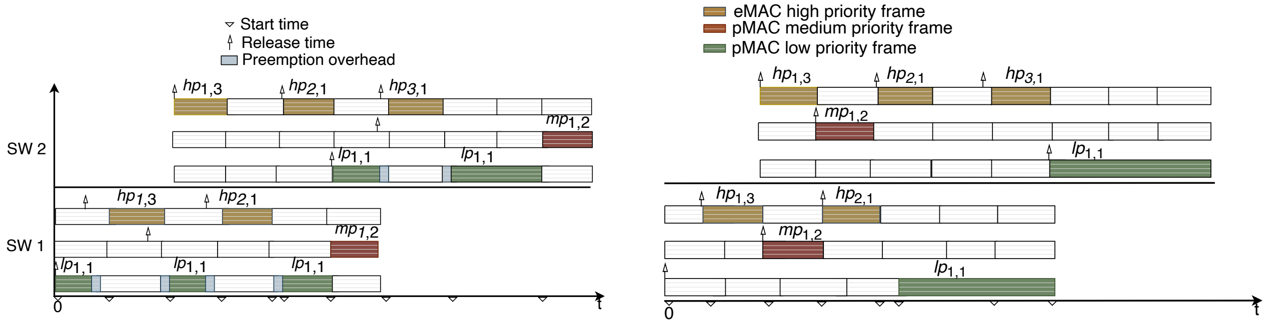
where $d_{i,j}$ is the absolute deadline of frame j in flow i, t is the current time, u is the time unit, $T_c = N \times u$ is the total cycle time, and N is the number of gates. Multiplying u by the number of stream gates gives the cyclical period during which the priority (i.e., the IPV) is changed. A cycle refers to a fixed time interval or period within which a frame is scheduled and transmitted. The cycle period or time determines the length of the cycle and the timing granularity of the network. After satisfying Eq. 1, the frame is scheduled for transmission using strict priority selection with the PCP derived by

$$PCP_{i,j} = n - 1 - \left\lfloor \frac{d_{i,j} - t - \tau_{bit}}{T_c} \right\rfloor \mod n. \tag{2}$$

C. Motivating Example

D-TSN ignores the potential for priority inversion in pMAC that is caused by scheduling eMAC frames according to their deadlines and ahead of the pMAC frames. As shown in Figure 2a, consider a network consisting of two switches (SW 1 and SW 2) that has two pMAC flows (lp and mp) and three eMAC flows (hp) with 1 frame each at the MAC layer. At $t = 0$, the lp frame arrives at the merge sublayer before the mp and first hp frame. While the lp frame is being transmitted, it gets preempted by an hp eMAC frame arriving at $t = 1$, and mp is released during the preemption. After the first hp frame completes, it moves to the link via switch 2 and lp resumes until another hp frame preempts it. Only after the lp frame finishes transmission at $t = 6$, can the mp frame transmit via SW 1. Concurrently, lp begins transmission via SW 2, but gets preempted by the arrival of a third hp frame (from some other node). mp finishes transmission and gets blocked at switch SW 2 again behind the preempted lp frame. Eventually, mp gets to transmit, but it can incur blocking by lower-priority frames at every switch. Priority inversion occurs when an mp frame must wait for a preempted lp frame to resume transmission after the hp frame has finished transmitting.

Our approach leverages the PCP and deadline-based scheduling to delay transmission of the lp frame at its source prior to release for transmission. This delay leads to a schedule as in Fig. 2b. The lp frame will not arrive in the MAC layer until after mp with this schedule because it is delayed by the EDF scheduler.



(a) Priority inversion in the preemption layer schedule.

(b) Schedule by delaying pMAC frame.

Fig. 2: Motivating Example. In 2a the medium-priority frame (mp) incurs blocking at multiple switches by the same low-priority frame (lp). Our approach manipulates the priority assignment to shift such low-priority frames later within the slack of their schedule to reduce the likelihood that such blocking can incur a deadline miss.

III. ILP Deadline-based TSN (ILP-DTSN)

In ILP-DTSN, we adopt Patti et al.'s D-TSN approach [7], inserting frames in transmission queues based on absolute deadlines. Our method involves online PCP selection by source nodes and offline service MAC assignment using ILP, comprising two components.

A. Priority Selection

Traditionally, the PCP is used to determine the priority of frames in TSN networks and therefore decide which frames get the higher priority queues, i.e., access to the eMAC interface. To ensure that only frames with close absolute deadlines are selected for transmission, we concentrate only on the calculation of PCP in the transmission configuration to assign priority to each frame. Then the eMAC/pMAC is allocated by considering the preemption cost, i.e., the worst-case delay a frame can experience due to preemption, using the ILP formulation in Section III-B.

Additionally, a configurable time unit u from Eq. 1 defines the limit of frames considered for transmission in a cycle as described in Section II-B. This limit, represented as $[d_a, d_b]$, guarantees that only frames with deadlines falling within this interval are selected for transmission. We denote the time that the i'th cycle ends as t_{c_i}. For example, if u is 20 μs, t_{c_3} would represent the time at the end of the third interval, which would be 60 μs. Priority inversion can occur when a low-priority (lp) frame arrives earlier than a medium-priority (mp) frame. In such cases, we modify the PCP in order to delay the transmission of the lp frame. Our approach delays the lower-priority frames unlike other models in which the higher-priority frames are delayed and multi-level preemption is used. Our approach for PCP selection is shown in Algorithm 1, where the PCP is denoted as $\rho_{i,j}$.

Generally, the interval from the arrival time $a_{i,j}$ of the j-th frame in the i-th flow plus its relative deadline D_i is known as the absolute deadline, and it is given by $d_{i,j} = a_{i,j} + D_i$, which is calculated as frames arrive, c.f. line 10. A frame's remaining time $RemTime$ (or slack) to the end of its deadline is calculated by deducting the current time t_{c_i} within the cycle period, which is calculated in line 9 as

Algorithm 1 Priority Selection

Output: List of priorities $\rho_{i,j}$ for all frames in F_i

1: **function** Priority Selection($a_{i,j}, F_i, Q_n, u, T_c$)
2: $\quad i \leftarrow 0$
3: $\quad t_{c_i} \leftarrow 0$
4: $\quad \rho_{\text{list}} \leftarrow []$
5: $\quad selected \leftarrow \{\}$
6: $\quad selected \leftarrow \{f_{i,j}$: False for $f_{i,j} \in F_i\}$
7: $\quad$ **while** $\exists\ f_{i,j}$ and $selected[f_{i,j}]$ is False **do**
8: $\quad\quad$ **for each** $f_{i,j} \in F_i$ **do**
9: $\quad\quad\quad d_{i,j} \leftarrow a_{i,j} + D_i$
10: $\quad\quad\quad RemTime \leftarrow d_{i,j} - t_{c_i}$
11: $\quad\quad\quad$ **if** $u < RemTime \leq T_c$ **then**
12: $\quad\quad\quad\quad [d_a, d_b] \leftarrow [d_{i,j} - u, d_{i,j}]$
13: $\quad\quad\quad\quad \rho_{i,j} \leftarrow n - 1 - \left\lfloor \frac{RemTime - Pr_{dl}}{T_c} \right\rfloor \mod n$
14: $\quad\quad\quad\quad selected[f_{i,j}] \leftarrow$ True
15: $\quad\quad\quad\quad \rho_{\text{list}}$.append($\rho_{i,j}$)
16: $\quad\quad\quad$ **end if**
17: $\quad\quad$ **end for**
18: $\quad\quad i \leftarrow i + 1$
19: $\quad\quad t_{c_i} \leftarrow t_{c_{i-1}} + u$
20: $\quad\quad$ **if** $a_{i,j} + t_{c_{i-1}} < t_{c_i}$ **then Selected is False**
21: $\quad\quad$ **end if**
22: $\quad$ **end while**
23: $\quad$ **return** ρ_{list}
24: **end function**

$$RemTime = d_{i,j} - t_{c_i}. \tag{3}$$

However, $RemTime$ should be less than or equal to the cycle time T_c so as not to miss the current cycle, which is checked by the conditional at line 11. The $selected$ set is used to keep track of frames that have already been considered for transmission. We then give a configurable deadline range of $[d_a, d_b]$ of frames to transmit within a cycle period. A frame is set to transmit if $d_{i,j} \in [d_a, d_b]$.

The calculation of $\rho_{i,j}$ is derived from the priority calculation of D-TSN [7], modified as follows:

$$\rho_{i,j} = n - 1 - \left\lfloor \frac{RemTime - Pr_{dl}}{T_c} \right\rfloor \mod n. \tag{4}$$

Here, Pr_{dl} accounts for preemption overhead, considering the potential delay of an express frame by up to 143 bytes (the length of the longest non-preemptable Ethernet frame fragment [9]). Additionally, each preemption incurs a total overhead of 24 bytes. We factor in preemption overhead once for each frame in our priority calculation. Also, we handle preemption overhead only for eMAC frames, with pMAC frames considered in the ILP model (refer to Section III-B). The preemption delay overhead is calculated as:

$$Pr_{dl} = max((143 \times 8\tau_{bit}), ((24 \times 8\tau_{bit}) + C_{i,j})) \tag{5}$$

where τ_{bit} represents the time to transmit one bit, and a conservative estimate is used by adding $C_{i,j}$ to account for blocking. A frame is identified as delayed at line 20 if, at the conclusion of the current cycle, $a_{i,j} + t_{c_{i-1}} < t_{c_i}$. Delayed frames undergo reconsideration through the loop iteration.

B. ILP Formulation

The performance objective of ILP-DTSN is to minimize the end-to-end delay for frames in each flow subject to constraints of the TSN standard and avoidance of deadline misses. With the constraints, all frames $f_{i,j}$ are scheduled based on their individual deadlines while ensuring their proper assignment to eMAC and pMAC queues denoted by α and β respectively. We define the objective function as:

$$Minimize \quad \sum_{i \in F} \sum_{j \in f} E_d \tag{6}$$

Where E_d is the end-to-end delay, such that: (i) The flows are schedulable (F_i deadline is not exceeded), (ii) Each frame with the closest deadline is assigned to the highest priority queue (iii) The queues with the closest deadline are assigned to the express service interface. In the following, we define these constraints formally.

1) Flow-to-Queue Constraints: Frames need to be selected for transmission based on their deadlines while avoiding collisions on the links and adequate resource availability along the frames' designated paths. In the following, we formulate each flow-to-queue constraint to achieve these requirements.

Routing Constraint: When frame $f_{i,j}$ of flow F_i selects the transmission route, the directed edges connecting the source node and the destination node must belong to the network. We model the network routes by binary path variables such that if F_i is routed through an edge from a node, the path variable $R_{i,j,l}$ is set to 1. Otherwise, it is set to 0, thus, we have:

$$R_{i,j,l} = \begin{cases} 1, & f_{i,j} \text{ is routed through } l. \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Frame Constraint: It is necessary that on all edges, different frames do not overlap. Thus, frames are scheduled so that the reserved times on any edge for two frames in a queue

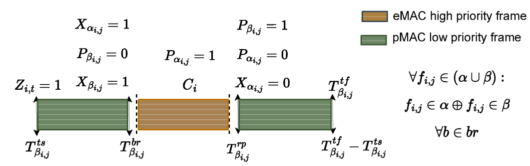


Fig. 3: Variables of Preemption Constraints.

do not overlap in every hyperperiod. We denote the start of transmission of a frame $f_{i,j}$ on link l as $t^{st}_{i,j,l}$ where T_i is the period and $C_{i,j,l}$ the transmission time of frame $f_{i,j}$ on link l. We can ensure that the reserved time slots for frames from different flows at a particular edge do not overlap with constraints:

$$\begin{aligned}
&\forall f_{i,j}, f_{i,k} \in F_i : \\
&\qquad f_{i,j} \neq f_{i,k} \\
&\qquad 0 \leq x < \frac{Hp}{T_i} \wedge 0 \leq y < \frac{Hp}{T_i}, \forall x, y \\
&\quad \text{if } R_{i,j,l} + R_{i,k,l} \geq 2 : \\
&\qquad t^{st}_{i,j,l} + x \cdot T_i - (t^{st}_{i,k,l} + y \cdot T_i) \geq C_{i,k,l} \\
&\qquad \quad \vee \\
&\qquad t^{st}_{i,k,l} + y \cdot T_i - (t^{st}_{i,j,l} + x \cdot T_i) \geq C_{i,j,l}
\end{aligned} \tag{8}$$

where H_p is the hyperperiod, $x \cdot T_i$ denotes the start of the x-th period for frame $f_{i,j}$ and $y \cdot T_i$ denotes the start of the y-th period for frame $f_{i,k}$. With these constraints, one of the overlapping frames must wait until the end of the transmission of the other frame before transmitting.

The selection of frames for transmission is based on their absolute deadlines, denoted as $d_{i,j}$, within a specified deadline range $[d_a, d_b]$. To derive an absolute deadline offline, the minimum of all absolute deadlines that arrive within the current cycle is found. To ensure that frames are selected based on their earliest deadlines, the following constraint is applied:

$$d_{i,j} = \min\{a_{i,j} + D_i : a_{i,j} \leq t_{c_i}\} \tag{9}$$

Queue Assignment Constraint: When the frame $f_{i,j}$ is transmitted through a switch it is buffered in the queue of the switch's egress port. This buffering leads to the following constraint:

$$\begin{aligned}
&\forall f_{i,j} \in F_i : \\
&\qquad a_{i,j} \leq t_{c_i} \\
&\qquad \rho_{i,j} \in \{0, \ldots, Q_n - 1\}
\end{aligned} \tag{10}$$

Each frame is assigned a PCP $\rho_{i,j}$ which is a function of its absolute deadline as calculated in Algorithm 1 and Eq. 4. The frame's PCP maps it to the corresponding queue. As before, since the exact arrival times are unknown this constraint imposes an inequality on arrival times within the current cycle.

2) Queue to eMac/pMAC Class Assignment Constraints: To incorporate the impact of preemption and preemption delay as a cost factor, we introduce the concept of the eMAC/pMAC queue constraint. This constraint considers the number of preemptions a frame may experience and evaluates the effect of preemption delay. Based on this evaluation, a decision is made to assign the frame to either the eMAC or pMAC class, as well as to determine the number of queues to allocate to the eMAC and pMAC.

Queue Class Assignment Constraint: Let $M = m_\alpha, m_\beta$ be the set of eMAC and pMAC interface, and Q_α, Q_β are subsets that partition the set of Q as eMAC and pMAC queues, respectively. We define binary variables e_q and p_q to represent the mapping of Q_α and Q_β to m_α and m_β, respectively, where $q \in (Q_\alpha \cup Q_\beta)$, $e_q = 1$ for all queues in m_α and $p_q = 1$ for all queues in m_β. Then we have

$$e_q = \begin{cases} 1, & q \in Q_\alpha \\ 0, & \text{otherwise} \end{cases} \qquad p_q = \begin{cases} 1, & q \in Q_\beta \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

Queues are mapped in priority order with all queues in Q_α (assigned to eMAC) having a greater index and therefore priority than queues in Q_β (assigned to pMAC). We enforce the constraint that $q_i \in Q$ is mapped to exactly one of the MAC queues as follows:

$$\begin{aligned} &\forall q \in Q_\alpha \cup Q_\beta : \\ &\qquad e_q + p_q = 1 \\ &\sum_{f_{i,j} \in F_i} e_{\rho_{i,j}} + p_{\rho_{i,j}} = 1 \end{aligned} \tag{12}$$

Furthermore, we ensure that the sum S of the sizes of all frames in Q_α, Q_β do not exceed the buffer sizes B_α, B_β allocated for eMAC and pMAC, respectively, by

$$\sum_{q \in Q_\alpha} S_q \leq B_\alpha \qquad \sum_{q \in Q_\beta} S_q \quad \leq B_\beta. \tag{13}$$

Transmission constraint: We keep track of frames that have been routed and transmitted to the MAC layer by time t by assigning to these frames a binary variable $Z_{i,t}$ given by:

$$Z_{i,t} = \begin{cases} 1, & f_{i,j} \text{ has been sent to the MAC layer.} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Preemption Constraints: When modeling preemption as an ILP problem, it is important to consider the interruption time of preempted frames and allocate the remaining time left for transmission when they resume. We are inspired by Castro et al. [10] who solve flowshop preemptive scheduling problems with break time using a continuous-time formulation. In our work, we allow other jobs to continue while a specific job is on break. This preemption model fits our problem reasonably well for the solver to make the expected decision in assigning which queue to use for the express or preemptable frame. Our problem is further constrained by the requirement that preemption will occur only if an express frame arrives while a preemptable frame is transmitted.

We introduce binary variables $P_{\alpha_{i,j}}$ and $P_{\beta_{i,j}}$ to represent the transmission of frame $f_{i,j}$ over the eMAC and pMAC respectively. We ensure the values taken by the variable $P_{m_{i,j}}$ are independent with respect to time slots. This means that frames in eMAC will not be preempted by pMAC frames, whereas pMAC frames can be stopped to resume transmission at another time slot. Therefore the constraint is given by:

$$\sum P_{\alpha_{i,j}} - \sum Z_{i,t} \leq 1 \tag{15}$$

The variable $Z_{i,t}$ represents the transmission decision for a frame at time t through the MAC layer. We furthermore ensure that at most one message transmits in any given time t with the following:

$$\begin{aligned} &\forall f_{i,j} \in F_i, t > 0 : \\ &\qquad Z_{i,t} \leq P_{\alpha_{i,j}} \\ &\qquad Z_{i,t} \leq P_{\beta_{i,j}} \\ &\qquad Z_{i,t} \geq P_{\alpha_{i,j}} + P_{\beta_{i,j}} - 1 \end{aligned} \tag{16}$$

A pMAC frame can transmit up to 128 bytes of the payload before being preempted by an available eMAC frame, according to the standard. Also from the standard, we expect each preemption overhead to cost (about) 12 bytes. We denote the start time in this phase as T^{ts}_β, break period as T^{br}_β, resumption period as T^{rp}_β, and finish time of such a preempted frame as T^{tf}_β. The pMAC frame is stopped from transmitting within the interval $[T^{br}_\beta, T^{rp}_\beta]$. The duration of this interval serves as the preemption cost $C_\beta = T^{rp}_\beta - T^{br}_\beta$. We add a constraint to ensure preempted frames can finish before their deadlines accounting for the extra preemption cost given by

$$T^{tf}_{\beta_{i,j}} \leq (T^{ts}_{\beta_{i,j}} + delay + C_i) P_{\beta_{i,j}} + \sum_{b \in br_\beta} C_\beta \leq d_{i,j} \tag{17}$$

For eMAC frames, from eqn 18, we consider the following constraint to meet their deadlines.

$$T^{tf}_{\alpha_{i,j}} \leq (T^{ts}_{\alpha_{i,j}} + delay + C_i) P_{\alpha_{i,j}} \leq d_{i,j} \tag{18}$$

Where $delay$ is the propagation and processing delay which is a (roughly) constant, network-specific value that includes the time it takes for a signal to propagate through the physical link between the nodes and the processing overhead at a switch.

We now introduce variables $X_{\alpha_{i,j}}$ and $X_{\beta_{i,j}}$ to denote an eMAC and pMAC frame, respectively, where the eMAC frame preempts the pMAC frame. Therefore, an eMAC frame is allowed to preempt a pMAC frame if the following holds:

$$\begin{aligned} X_{\alpha_{i,j}} &= 1 \\ X_{\beta_{i,j}} &= 1 \\ P_{\alpha_{i,j}} T^{ts}_{\alpha_{i,j}} + (128 \times 8)\tau_{bit} \leq a_{i,j} &\leq P_{\beta_{i,j}} T^{ts}_{\beta_{i,j}} + C_k \end{aligned} \tag{19}$$

The constraint in Eq. 19 ensures that a pMAC frame is preempted at any time an eMAC frame arrives within a period when the pMAC is chosen to transmit plus its transmission time. Further, we also allow the preemption to occur after the pMAC frame has transmitted up to 128 bytes.

We allow the transmission of the eMAC frame to commence by ensuring that the pMAC frame observes the break period. To do that, we consider $T^{ts}_{\alpha_{i,j}}$, the time eMAC frame ($X_{\alpha_{i,j}}$) will be chosen to transmit. The binary variable that allows the eMAC/pMAC frame to transmit after queuing is $Z_{i,t}$ = 1. The frame scheduling constraint guarantees that no two frames can transmit simultaneously at the MAC layer. When the pMAC frame is preempted, from Eq. 19, $X_{\alpha_{i,j}} = 1$. Then, we ensure that the eMAC frame gets transmitted only after the blocking time (break period) of the pMAC frame. This break period will be less than or equal to the start of the eMAC frame transmission plus its transmission time. The time eMAC frame finishes transmitting should be less than or equal to the resumption of the preempted pMAC frame. The pMAC frame then continues to its finish time or experiences another preemption. The constraint is given by:

$$\begin{aligned} T^{br}_{\beta_{i,j}} &\leq T^{ts}_{\alpha_{i,j}} + C_i \\ T^{tf}_{\alpha_{i.j}} &\leq T^{rp}_{\beta_{i,j}} \leq T^{tf}_{\beta_{i,j}} Z_{i,t} \end{aligned} \tag{20}$$

Since from Eq. 15, $P_{\beta_{i,j}}$ must have been changed to 1 to transmit, then the below constraint helps to change $P_{\beta_{i,j}}$ back to 0 for the eMAC frame to transmit. This model is used for all the frames until completion.

$$\begin{aligned} P_{\alpha_{i,j}} + P_{\beta_{i,j}} &= 1 \\ P_{\alpha_{i,j}} X_{\alpha_{i,j}} + P_{\beta_{i,j}} (X_{\beta_{i,j}} - 1) &= 1 \end{aligned} \tag{21}$$

Finally, after the break period of pMAC frames and there are no other eMAC frames to transmit. Eq. 21 becomes

$$\begin{aligned} P_{\alpha_{i,j}} + P_{\beta_{i,j}} &= 1 \\ P_{\alpha_{i,j}} (X_{\alpha_{i,j}} - 1) + P_{\beta_{i,j}} X_{\beta_{i,j}} &= 1 \end{aligned} \tag{22}$$

IV. Evaluation

We implemented the ILP with the specified constraints using Gurobi, a Python-based optimizer solver. We simulate the behavior of the network devices, including the switches, using SimPy, a discrete event simulator. We designed a network application in SimPy to perform inter-layer optimization and routing of frames between nodes. We integrated the ILP formulation into the SimPy model to enhance the traffic routing and scheduling while ensuring adherence to the constraints outlined in Section III-B. We used the OMNeT++ framework along with the Nesting simulator, which supports physical layer preemption as defined in the IEEE 802.1Qbu standard [11] to evaluate and validate the results. The number of eMAC/pMAC queues, the flows that are mapped to queues, and the routing of each flow all are used to configure the Nesting framework.

A. Experiment 1: Comparison of ILP-DTSN with TAS

In this experiment, we compare the end-to-end delay of TAS with ILP-DTSN for a 2-switch, 8-node topology as used previously by Castro et al. [10], which simulates an in-vehicle network that handles the communications of Advanced Driver Assistance Systems (ADAS) and multimedia/infotainment with frame preemption enabled. The network operates at a data rate of $100Mbps$. The processing delay of each switch is set to $1\mu s$, and the propagation delay is set to $100ns$. We adapted the flow parameters with slight changes from the original [10] for this experiment, which are presented in Table II. We set the time interval $u = 120$, slightly above the minimum deadline in the flows. The cycle time is set at $n \times u$ in this case, and Algorithm 1 is used to determine the values $[d_a, d_b]$. At the end of each time slot, we increase these values by $120\mu s$. The frame priority is calculated based on the flow's deadline and assigned to the VLAN Tag of the frame. Flows with shorter deadlines are assigned to queue 7 within a cycle, while the best-effort flows had to wait until $t_{c_{i=74}}$ before being transmitted. This approach ensures that frames with higher deadlines, such as frames from $f_{1,1}$ are transmitted when $t_{c_{i=0}}$, to avoid missing their deadline.

TABLE II: Experiment 1 ADAS Flow Parameters [10].

Flow ID	P(μs)	Src	Size(B)	Dst
1	100	ES_1	46	ES_4
2	200	ES_1	46	ES_4
3	200	ES_5	184	ES_3
4	100	ES_5	184	ES_2
5	200	ES_5	184	ES_2
6	100	ES_4	184	ES_5
7	500	ES_4	400	ES_6
8	1000	ES_4	718	ES_6
9	500	ES_4	600	ES_7
10	1000	ES_4	800	ES_7
11	1000	ES_6	500	ES_1
12-14	100	ES_8	80	ES_4
15	200	ES_8	350	ES_4
16-17	10000	$ES_{2,3}$	1496	ES_8
18	10000	ES_7	1496	ES_5

We compare ILP-DTSN as described in Section III with TAS. With TAS, each flow has a pre-defined traffic class. According to the standard [12], class A and class B may be assigned priority values of 3 and 2, respectively. We compare our ILP formulation against the ILP TAS proposed by Hellmans et al. [13]. The TAS scheduling depends on the cycle time as opposed to the deadline scheduling that ILP-DTSN uses. The main constraint considered in the TAS scheduling is to ensure that the completion time for a flow does not exceed the cycle time. However, this constraint does not take into account the deadline and can lead to increased blocking delays when different traffic pass through.

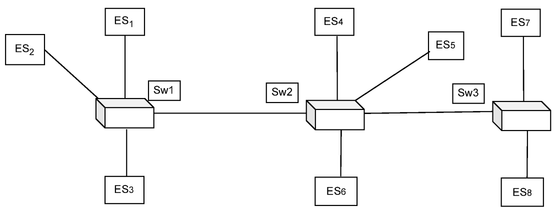
Table III shows the average end-to-end delay with the described parameters. The result demonstrates a minimized end-to-end delay with our model compared to the TAS configuration, where blocking delays were experienced by flow 4 due to preempting best-effort traffic. On average, flow 4 exceeded its deadline with TAS. In this example, ILP-DTSN decides to use only queue 7 as the eMAC queue, and it delayed frames from flows 16-18 by giving them a priority of 2. This configuration reduces the network load due to the delayed flows, benefiting flows like F_4.

B. Experiment 2: Flows without Best-Effort Traffic

We also conducted an evaluation of our approach on flows without any best-effort traffic and using a 3-switch topology as depicted in Fig. 4a. In this case, the TAS traffic was scheduled with different classes. The properties of the flows are presented in Table 4b. Class A traffic is considered as medium priority frames, which are subjected to delays from class B traffic in the same preemptable queues. In this experiment, we selected

TABLE III: Experiment 1 average end-to-end delay.

Flow ID	Deadline	Class	ILP-DTSN	TAS
1	100	ST	17.3	17.3
2	200	ST	18.5	24.4
3	200	A	62.4	81.9
4	100	A	66	100.7
5	200	A	102	118.7
6	100	A	62.8	62.8
7	500	B	131.5	131.5
8	1000	B	311.2	366.3
9	500	B	119.2	187.4
10	1000	B	221	373.5
11	1000	B	197.2	280
12-14	100	A	49	56.2
15	200	B	74.9	95.3
16-17	-	BE	-	-
18	-	BE	-	-



(a) 3-switch topology

Flow ID	Src	Dst	Class
1	ES_1	ES_7	A
2	ES_2	ES_7	ST
3	ES_4	ES_7	ST
4	ES_5	ES_7	A
5	ES_6	ES_7	B
6	ES_6	ES_8	ST
7	ES_8	ES_7	B
8	ES_6	ES_8	ST
9	ES_6	ES_8	A
10	ES_8	ES_7	B

(b) Flow Parameters for 3-switch network

Fig. 4: Experiment 2 Topology and Flow Parameters.

periods from the set $\{1000, 5000, 10000\}$ μs and calculated the resulting payload up to 1500B using a data rate of $100 Mbit/s$. We averaged the end-to-end delay from 15 stream traffic frames and compared the results of TAS with ILP-DTSN. For this case, the cycle time u was set to 500 μs.

The results shown in Fig. 5a indicate that ILP-DTSN reduces delays for ST traffic. The highest reduction was when the TAS ST shows 960 μs and ILP-DTSN shows 650 μs. This is expected as the frames with higher deadlines are delayed for the lower-priority frames, hence, decreasing their overall end-to-end delay. For the class A traffic we see more improvement as—unlike the fixed lower priority queue used in the TAS approach—some of these frames were assigned to higher priority queues. However, the class B frames suffer more when we delay them in most cases as shown in Fig. 5c.

In addition to evaluating the average end-to-end delay, we investigated the maximum delay using periods of $500, 1000, 1500$ μs with u set to 32. The results, illustrated in Fig. 6, compare TAS with ILP-DTSN. In scenarios of high network utilization, the maximum delay may exceed the period, resulting in deadline misses. Notably, ILP-DTSN demonstrates consistently lower maximum end-to-end delays at lower and intermediate network utilization levels than TAS.

TABLE IV: Flow Parameters from Multi-Level Preemption [5]

F	P	D	Src	Size	Dst
1	500	120	ES_1	300	ES_4
2	700	250	ES_2	700	ES_5
3	100000	-	ES_3	1500	ES_6

TABLE V: Comparison of ILP-DTSN with multi-level preemption [5] and one-level preemption using TAS.

F	**Worst-case E2E-Delay**		
	TAS	Multi-Level	ILP-DTSN
1	109	109	109
2	253	203	186
3	-	-	-

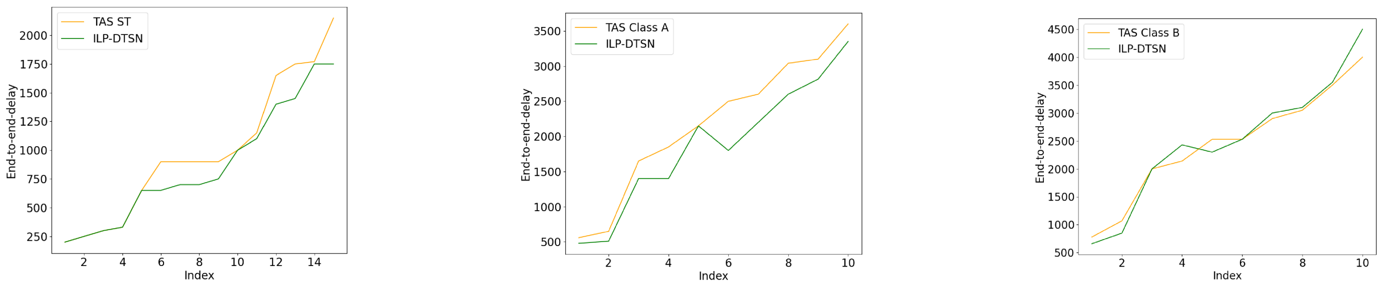
C. Experiment 3: Comparison with Multi-Level Preemption

We compare ILP-DTSN and the multi-level preemption mechanism proposed by Ojewale et al. [5] reusing their topology and flow parameters, shown in Table IV. The evaluation focused on the end-to-end delay of three flows: F_1, F_2, and F_3, as depicted in Table IV. ILP-DTSN assigned F_1 to queue 7 and F_2 to queue 6. Both flows were scheduled to transmit at time $t = 0$. F_3 was scheduled to transmit at $t = 2$ ms to avoid delays in the network.

Table V depicts the worst end-to-end delay of flows F_1 and F_2 comparing one-level preemption using ILP-DTSN, one-level preemption, and multi-level preemption [5]. Note that the values for multi-level preemption are reproduced from their paper, as we were only able to reproduce their approach in part due to the differences in the network architecture. However, based on the identical performance we obtained for flow F_1, we are confident that our results are comparable. When comparing the results of ILP-DTSN with the multi-level preemption mechanism for flow F_2, we observed a lower overhead delay of approximately 9%. F_2 encountered no blocking from F_3 instances and did not undergo an additional layer of preemption. Note that the delay reduction achieved in this specific simulation did not consider any additional overhead that may arise from architectural changes required by the multi-preemption approach. We expect the performance of ILP-DTSN would be significantly improved with multiple switches and many best-effort frames.

V. Related Work

Park et al. [14] discussed the synthesis problem for assigning frames to eMAC and pMAC and proposed a genetic algorithm as a heuristic solution but do not consider the entire traffic flow or its effects on deadline misses. Lo Bello et al. [15] proposed a schedulability analysis for TSN with scheduled traffic and preemption support focusing on the credit-based shaper and TAS, whereas our work addresses the priority inversion problem with deadline-based TSN and frame preemption. Ojewale et al. [5] identified the problem of priority inversion for preemptable traffic with timing requirements and



(a) Comparison with TAS ST (b) Comparison with TAS Class A (c) Comparison with TAS Class B

Fig. 5: End-to-end delay of 15 frames with different configurations.

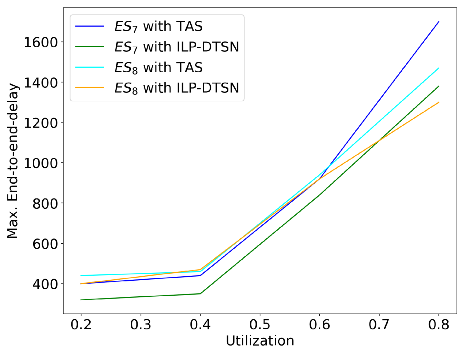


Fig. 6: Maximum E2E delay for all flows.

proposed multi-level preemption, which modifies the IEEE 802.1Qbu standard and results in architecture modifications and higher preemption overhead. Ashjaei et al. [6] refined multi-level preemption with a nested preemption approach that uses a credit-based shaper to schedule frames, but still does not follow the standard because pMAC frames are allowed to preempt each other. Zou et al. [16] discussed the problems of preemption with the credit-based shaper and suggested a constant bandwidth server using multi-level preemption that allows eMAC frames to be preempted by the pMAC frame; their approach does not follow the standard and introduces overhead. Patti et al. [7] proposed deadline-based TSN model (D-TSN) but did not consider frame preemption in the MAC layer. In contrast to the prior work, our approach considers frame preemption, scheduling, and routing in the network while adhering to TSN standards' requirements.

VI. Conclusion

ILP-DTSN efficiently allocates frames to eMAC and pMAC service interfaces of TSN switches. Our approach considers frame preemption when assigning priority and incorporates a joint scheduling and routing procedure to address the priority inversion issue in TSN while ensuring network QoS. We have shown that ILP-DTSN can reduce end-to-end delay up to 32% without the need for an extra layer of preemption. Our approach reduces the likelihood of time-sensitive traffic missing their deadlines and enhances overall network performance.

Acknowledgements

This work is partially supported by NSF CNS-2046705, NSF ECCS-2138295, and Colorado State Bill 18-086.

References

[1] J.-L. Scharbarg, M. Boyer, and C. Fraboul, "Can-ethernet architectures for real-time applications," in *2005 IEEE Conference on Emerging Technologies and Factory Automation*, vol. 2. IEEE, 2005, pp. 8–pp.

[2] "IEEE std 802.1q-2018: IEEE standard for local and metropolitan area networks – bridges and bridged networks," IEEE Standards Association, Dec 2018.

[3] "Ieee standard for local and metropolitan area network–bridges and bridged networks," *IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014)*, pp. 1–1993, 2018.

[4] "Ieee standard for ethernet amendment 5: Specification and management parameters for interspersing express traffic," *IEEE Std 802.3br-2016*, pp. 1–58, 2016.

[5] M. A. Ojewale, P. M. Yomsi, and B. Nikolić, "Multi-level preemption in tsn: feasibility and requirements analysis," in *2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC)*. IEEE, 2020, pp. 47–55.

[6] M. Ashjaei, M. Sjödin, and S. Mubeen, "A novel frame preemption model in tsn networks," *Journal of Systems Architecture*, vol. 116, p. 102037, 2021.

[7] G. Patti, L. Lo Bello, and L. Leonardi, "Deadline-aware online scheduling of tsn flows for automotive applications," *IEEE Transactions on Industrial Informatics*, 2022.

[8] F. R. Abdalla Mahamid and K. Holzinger, ""time sensitive networking - 802.1qci"," https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2021-05-1/NET-2021-05-1_03.pdf, accessed: 2022-12-18.

[9] D. Thiele and R. Ernst, "Formal worst-case performance analysis of time-sensitive ethernet with frame preemption," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2016, pp. 1–9.

[10] P. M. Castro, I. Harjunkoski, and I. E. Grossmann, "Discrete and continuous-time formulations for dealing with break periods: Preemptive and non-preemptive scheduling," *European Journal of Operational Research*, vol. 278, no. 2, pp. 563–577, 2019.

[11] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer, and K. Rothermel, "Nesting: Simulating ieee time-sensitive networking (tsn) in omnet++," in *2019 International Conference on Networked Systems (NetSys)*. IEEE, 2019, pp. 1–8.

[12] IEEE, "Ieee standard for local and metropolitan area networks–timing and synchronization for time-sensitive applications," https://standards.ieee.org/ieee/802.1AS/7121/, 2020, accessed: 2022-12-18.

[13] D. Hellmanns, L. Haug, M. Hildebrand, F. Dürr, S. Kehrer, and R. Hummen, "How to optimize joint routing and scheduling models for tsn using integer linear programming," in *29th International Conference on Real-Time Networks and Systems*, 2021, pp. 100–111.

[14] T. Park, S. Samii, and K. G. Shin, "Design optimization of frame preemption in real-time switched ethernet," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 420–425.

[15] L. Lo Bello, M. Ashjaei, G. Patti, and M. Behnam, "Schedulability analysis of time-sensitive networks with scheduled traffic and preemption support," *Journal of Parallel and Distributed Computing*, vol. 144, pp. 153–171, 2020.

[16] J. Zou, X. Dai, and J. A. McDermid, "retsn: Resilient and efficient time-sensitive network for automotive in-vehicle communication," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.