TSN Schedule Evaluation for Online Compression

Fabian Kummer, Michael Nast, Frank Golatowski, Christian Haubelt
Institute of Applied Microelectronics and Computer Engineering
University of Rostock
18051 Rostock, Germany
firstname.lastname@uni-rostock.de

Willi Brekenfelder, Helge Parzyjegla, Peter Danielis
Institute of Computer Science
University of Rostock
18051 Rostock, Germany
firstname.lastname@uni-rostock.de

***Abstract*—In Time-Sensitive Networking (TSN), Time-Aware Shaping facilitates the convergence of deterministic and low-priority traffic in Ethernet networks. The allocation of bandwidth for low-priority traffic is crucial when scheduling time-triggered (TT) streams to prevent starvation and packet loss caused by queue congestion. This issue is particularly pertinent in online scheduling scenarios, where the deletion of streams can lead to schedule fragmentation. To prevent this issue, the time slots of the TT streams must be compressed to provide larger contiguous gaps for low-priority traffic in the schedule. In this paper, we present a novel heuristic approach for compressing schedules, which facilitates the grouping of time slots into units, thereby increasing the bandwidth for low-priority traffic and improving the schedulability of future streams. The proposed heuristic can be tailored to specific use cases through adaptive weighting, allowing a comprehensive evaluation of time slots according to the properties of the associated time slot units, TT streams, and the network.**

***Index Terms*—Online reconfiguration, schedule compression, time-sensitive networking (TSN).**

I. Introduction

The increasing demand for real-time communications in various domains, including industrial automation, automotive, and medical applications, has emphasized the need for highly deterministic networks capable of meeting stringent latency and jitter requirements and facilitating the coexistence of time-triggered (TT) and low-priority traffic in dynamically evolving networks. In order to address these requirements in Ethernet networks, a series of standards for Time-Sensitive Networking (TSN) [1] have been established. TSN extends Ethernet by integrating mechanisms that enable precise time synchronization, guarantee deterministic forwarding, and frame isolation of data frames from TT streams. Additionally, it facilitates the reservation of bandwidth for rate-constrained (RC) traffic and the integration of best-effort (BE) traffic.

Conventional TSN scheduling approaches typically rely on static schedules that are computed offline and configured during network deployment. However, this inflexibility renders them unsuitable for use cases that necessitate dynamic adaptation at runtime, such as integrating new devices into the network topology or integrating/deleting streams. Dynamic adaptation of existing schedules is crucial for these use cases, but it can lead to schedule fragmentation, resulting in small gaps between time slots, for example when a stream is deleted. These gaps are filled with guard bands and limit the flexibility to integrate new streams in the future.

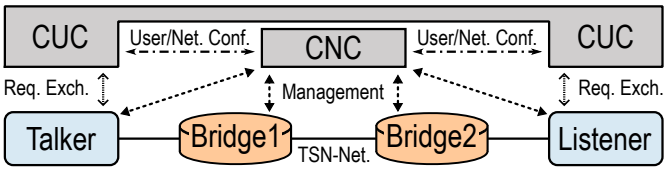


Figure 1. Fully Centralized TSN Configuration Model [1].

To address this challenge, we propose a schedule evaluation heuristic that considers the requirements of time-triggered and low-priority traffic in order to compress the time slots in a schedule. The objective of this heuristic is to prevent the starvation of low-priority traffic in time-sensitive networks and improve the schedulability for future stream integration. Our approach conducts an evaluation of individual time slots, considering time slot grouping, gaps between time slots, stream reconfigurability without path alteration, link utilization, and minimization of stream latency. For these criteria, we define evaluation functions that can be individually adjusted and weighted to form a unified assessment metric.

The remainder of this paper is structured as follows: The network architecture and the concept of scheduling in TSN are presented in Section II. In Section III, we introduce the heuristic for evaluating time slots with regard to schedule compression. We discuss related work in Section IV. A concluding summary of our work and implications for future investigations is provided in Section V.

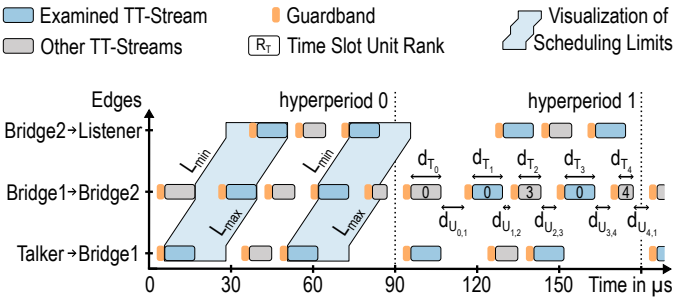
II. Scheduling in TSN

The scheduling of TT streams necessitates the implementation of the Fully Centralized TSN Configuration Model [1], as illustrated in Figure 1. This model consists of bridges that forward frames, as well as endpoints that generate data traffic (Talker) or consume it (Listener). In online scenarios, stream requirements are centrally collected at the Centralized-User-Configuration (CUC) and then transmitted to the Centralized-Network-Controller (CNC), which calculates a schedule for all stream and implements the configuration on the network devices.

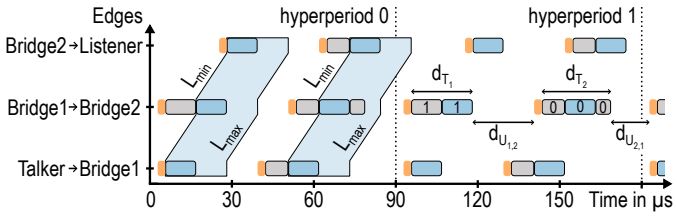
Scheduling in TSN utilize a topology G consisting of nodes N and directed edges E that abstracts the TSN network. The notation for these elements is specified in Table I. The directed edges E represent the outgoing interfaces that enable

Table I
FORMULA NOTATION

Notation	Definition	Attributes
N	node	
N_S	source node	
N_D	destination node	
E	outgoing edges	$E=(id,N_S,N_D)$
G	topology	$G=(\{N\},\{E\})$
t_B	base time	
p_H	hyperperiod	
n_H	hyperperiod number	
t_H	hyperperiod start time	$t_H=t_B+p_H\cdot n_H$
t_S	stream offset	
p_S	stream period	$p_S \mid p_H$
n_S	stream period number	
t_{Sn}	stream period start time	$t_{Sn}(n)=t_H+t_S+p_S\cdot n_S$
D_S	relative stream deadline	
r_S	stream path N_S to $\{N_D\}$	$p_S=\{E\}$
L	stream latency	
l_S	frame size (layer 1)	
q	queue assignment	
T	time slot	$T=(t_{\text{Start}},t_{\text{End}},q)$
T_S	stream time slot allocation	$T_S=\{T\}$
d_T	time slot unit duration	
d_U	unoccupied duration	
S_{req}	stream requirements	$S_{req}=(p_S,D_S,l_S,N_S,\{N_D\})$
S	scheduled TT-stream	$S=(t_S,p_S,D_S,r_S,l_S,T_S)$
scd	schedule	$scd=\{S\}$



(a) Uncompressed TSN schedule for time-triggered streams.



(b) Compressed TSN schedule for time-triggered streams.

Figure 2. Exemplary schedule based on the topology of Figure 1, with labeled parameters for evaluating the time slots, d_{T_x} = time slot unit duration, $d_{U_{x,y}}$ = unoccupied duration between time slot units, L_{min} = minimal latency on shortest path from source N_S to node N, L_{max} = maximum latency from node N to reach destination N_D within deadline on shortest path.

the time-triggered frame forwarding. Based on a topology, scheduling can be performed for a set of stream requirements $\{S_{req}\}$, either offline prior to runtime or online during runtime. Stream requirements S_{req} include a period p_S, a relative deadline D_S, a frame size l_S (single frame or burst size), as well as a source node N_S and one or more destination nodes N_D. During scheduling, an offset t_S relative to the hyperperiod start time t_H is assigned to each data stream S, and time slots T_S are allocated along a path r_S.

III. SCHEDULE COMPRESSION HEURISTIC

Figure 2 represents a schedule based on the topology shown in Figure 1. For the outgoing edges of the topology, the time progression of the time slots of two hyperperiods is shown. In online scheduling, where streams can be integrated, modified, and deleted during operation, schedule fragmentation may occur, as exemplified in Figure 2a. This fragmentation can arise due to various factors, primarily when changes are made to the schedule without adjusting the remaining running streams. Deleting an existing stream can create a gap in the schedule, especially if it was previously allocated in a unit of time slots. When integrating a new stream with a short deadline, it may not be possible to place the time slots on the edges of units, but only in between, which also results in additional shorter gaps.

These fragmented schedules, as shown in Figure 2a, where time slots are arranged individually or in small groups rather than as a contiguous units, can lead to problems in future scheduling and integration of low-priority traffic such as RC and BE traffic. The future scheduling process could become more challenging due to the prevalence of shorter unoccupied time slots, as potentially fewer possible stream configurations can be found. Consequently, it becomes necessary to reschedule existing streams before integrating new ones, which can also increase the computational time for scheduling, potentially leading to problems at time-critical integrations. To ensure frame isolation of TT streams it is necessary to set a guard band before each time slot unit, which completely blocks the edge to protect TT traffic from interference of low-priority traffic. With increasing number of time slot units, the available bandwidth for low-priority traffic decreases.

To mitigate these issues, the property of real-time traffic being valid as long as the deadline is met can be leveraged, along with the potential queuing in outgoing interfaces. To enhance the future schedulability of the schedule and to increase the bandwidth for low-priority traffic, the schedule should be compressed by rescheduling time slots of TT streams into seamless time slot units, as shown in Figure 2b. This compression can occur together with scheduling or separately after a new schedule has been configured in the network. Separation has the advantage of reducing the time until a new stream is integrated into the network. To achieve schedule compression, an evaluation of the schedule and its time slots should be conducted. Depending on the requirements for the schedule, different evaluation criteria may emerge.

A. Time Slot Placement

In order to compress the schedule, the already scheduled time slots should be evaluated individually. The requirements imposed on the schedule can lead to different evaluation criteria. In this work, we define several evaluation criteria for a heuristic to assess schedules and time slots, which can be adapted to different application scenarios. As illustrated in Figure 2 within hyperperiod 1, a crucial evaluation criterion is the duration d_T of a time slot unit. A high duration of time slot units leads

to better schedule compression. However, an excessively large duration can cause starvation of low-priority traffic, which may manifest as prolonged queuing times or packet loss due to congested queues. Therefore, finding a compromise between these two opposing effects of time slot duration d_T is essential.

$$\sigma(x,a,b) = \frac{1}{1+e^{-\frac{x-a}{b}}}$$
$$\sigma_{rev}(x,a,b) = \frac{1}{1+e^{\frac{x-a}{b}}}$$

For the evaluation, we use a sigmoid function σ, which represents an input value according to a nonlinear monotonically increasing function with costs in the value range $[0,1]$. In addition to this monotonically increasing sigmoid function σ, we also employ a monotonically decreasing sigmoid function σ_{rev}. To adjust σ and σ_{rev}, we use the parameters a, which specifies the turning point at a cost of 0.5 and b, which determines the slope of the curve. Besides the sigmoid function, other functions with a value range $[0,1]$ can be used, such as the heaviside function.

$$c_{T_{min}}(d_T) = \sigma_{rev}(\frac{d_T}{p_H}, a_{T_{min}}, b_{T_{min}})$$
$$c_{T_{max}}(d_T) = \sigma(\frac{d_T}{p_H}, a_{T_{max}}, b_{T_{max}})$$

We evaluate the minimum and maximum durations of time slot unit duration d_T separately, to account for different requirements for schedule compression and starvation. The evaluation of the minimum time slot unit duration is performed by $c_{T_{min}}$, where high costs are incurred until the minimum desired duration $a_{T_{min}}$ is reached, after which the costs decrease exponentially. Conversely, to avoid starvation of low-priority traffic, we use $c_{T_{max}}$ to evaluate the maximum time slot unit duration. The cost remains low until the maximum desired duration $a_{T_{max}}$ is reached, after which it increases exponentially.

$$c_R(R_T) = \sigma(R_T, a_R, b_R)$$

In addition to evaluating time slot unit duration d_T, we also evaluate the rank R_T of the associated time slot unit on the edge E for each time slot. This evaluation is performed analogously to a ranking list, as depicted in Figure 2 within hyperperiod 1. We do this because few but long time slot units are beneficial for schedule compression and guard band reduction. The evaluation of the rank is also conducted using the sigmoid function c_R, where the desired maximum number of time slot units can be adjusted with a_R. All time slots with a higher rank than the desired value are evaluated with high costs.

$$c_U(d_{U_x}, d_{U_y}) = \sigma_{rev}(\frac{\min\{d_{U_x}, d_{U_y}\}}{p_H}, a_U, b_U)$$

Besides the direct evaluation of the time slot units, the unoccupied duration d_U between time slot units is crucial, as shown in Figure 2 within hyperperiod 1. If the unoccupied duration between time slot units is too short, low-priority traffic may still starve. To evaluate the unoccupied duration to the neighboring time slot units, we use c_U, where the shorter duration is considered relative to the hyperperiod. The desired minimum unoccupied duration can be set with a_U.

B. Flexibility and Latency

In addition to evaluating time slot units, we also evaluate time slots individually based on the characteristics of the associated stream. The properties of a scheduled TT stream S, as defined in Table I, facilitate the calculation of a minimum latency L_{min}, which corresponds to the latency along the shortest path from the source N_S to a certain node without queuing. Conversely, the minimum latency from a certain node to the destination N_D can also be utilized by subtracting it from the deadline D_S to determine the maximum latency L_{max}. The maximum latency L_{max} represents the highest permitted latency at a certain node to ensure that the destination N_D is reached within the deadline D_S. Scheduling time slots is only permissible within the bounds defined by L_{min} and L_{max}, as shown in Figure 2 within hyperperiod 0.

Rescheduling running streams is essential to compress a fragmented schedule. In this process, shifting time slots along the consisting path is preferable, as time-triggered switching of paths is currently not feasible with TSN. To evaluate the feasibility of shifting a stream along its assigned path, we use the measure of flexibility F as defined in [2].

$$Free(t,E) = \begin{cases} 1, \text{ if unoccupied time slot on E} \\ 0, \text{ if blocked by time slot on E} \end{cases}$$
$$F_T(S,n_S,E) = \int_{t_S+n_S p_S+L_{\min}(S,E)}^{t_S+n_S p_S+L_{\max}(S,E)} Free(t,E)dt$$

The flexibility of a time slot F_T is calculated on an outgoing edge E for the associated scheduled stream S at a specific period number n_S and corresponds to the unoccupied duration d_U within the interval between L_{min} and L_{max} on the edge. The function $Free$ determines whether an edge is occupied or unoccupied at a given point in time. By integrating over the interval between the minimum and maximum latency of the period n_S, the unoccupied duration d_U in the interval corresponding to the flexibility F_T is obtained. This estimation does not take into account guard bands or possible time slot configurations, but allows an efficient evaluation of a stream without the need for recursive evaluation of other streams.

$$c_{F_T}(S,n_S,E) = \sigma_{rev}(\frac{F_T(S,n_S,E)}{p_H}, a_F, b_F)$$

We evaluate the flexibility of a time slot F_T in relation to the hyperperiod p_H and derive a cost value c_{F_T} using a sigmoid function σ_{rev}. By adjusting the turning point a_F, the minimum desired ratio of unoccupied duration can be configured.

$$c_{F_n}(S,n_S) = \max_{E \in r_S} \{c_{F_T}(S,n_S,E)\}$$
$$c_F(S) = \max_{n_S \in [0, p_H/p_S] \cap \mathbb{Z}} \{c_{F_n}(S,n_S)\}$$

Since the scheduling of a time slot is stream path dependent, we also consider the flexibility of the stream period c_{F_n}. This is determined by taking the maximum cost value, which corresponds to the minimum flexibility along the entire path. If independent scheduling of individual stream periods within a hyperperiod is permitted, then subsequent rescheduling of time slots can also be performed for individual stream periods. Therefore,

the flexibility for the period c_{F_n} is sufficient to evaluate the stream. If the individual planning of periods is prohibited, the total stream flexibility c_F must be evaluated by the maximum of all stream periods n_S of the scheduled stream S.

$$c_{EU}(E) = \sigma_{rev}\left(\int_0^{p_H} Free(t,E)\frac{dt}{p_H}, a_{EU}, b_{EU}\right)$$

In addition to the flexibility, we also evaluate the edge utilization using the sigmoid function c_{EU}. By adjusting a_{EU}, the maximum desired utilization for individual links can be configured. This approach facilitates the rescheduling of scheduled streams away from highly utilized to less congested paths.

$$c_{L_T}(S,n_S,E) = \left.\frac{L(S,n_S,E) - L_{min}}{L_{max} - L_{min}}\right|_{L_{max} > L_{min}}$$

Finally, we incorporate the stream latency into the evaluation by considering, at each node, the ratio of the latency L to L_{max}, adjusted by subtracting L_{min}. The resulting ratio serves as a cost function c_{L_T} to evaluate the streams latency at a certain node. The objective of this evaluation is to optimize for low latencies during schedule compression, which, in turn, leads to reduced queue utilization and promotes the use of the shortest path.

C. Weighted Time Slot Rating

To determine a cost value, we define a cost vector $\vec{c}$ for each time slot, which is composed of the cost functions described in Section III-A and Section III-B.

$$\vec{c} = (c_{T_{min}}, c_{T_{max}}, c_R, c_U, c_{F_T}, c_{F_n}, c_F, c_{EU}, c_{L_T})$$
$$\vec{w} = (w_{T_{min}}, w_{T_{max}}, w_R, w_U, w_{F_T}, w_{F_n}, w_F, c_{EU}, w_{L_T})$$
$$c = \vec{c} \cdot \vec{w}^T = c_{T_{min}} \cdot w_{T_{min}} + \cdots + c_{L_T} \cdot w_{L_T}$$

A weight vector $\vec{w}$ is defined to weigh the individual costs, assigning a weight in the interval $[0,1]$ to each cost function, with the sum of the weights being normalized to $\|\vec{w}\|_1 = 1$. By computing the scalar product of $\vec{c}$ and $\vec{w}$, a cost value c for a time slot can be calculated, which also exhibits in the interval $[0,1]$ due to the constraints imposed by $\vec{c}$ and $\vec{w}$. The higher the cost of a time slot, the worse its position in the schedule and the more advantageous it is to reschedule it.

The weighting with $\vec{w}$ allows for a fine-grained adaptation to specific requirements, as certain cost factors can be assigned higher or lower weights or even deactivated entirely. For instance, if individual stream period scheduling is permitted, it becomes unnecessary to consider the total stream flexibility, so the weight w_F can be set to 0 to deactivate the costs c_F.

This heuristic can be utilized to determine the order of reconfiguring streams to obtain a compressed schedule. Additionally, it can be employed to establish a threshold beyond which it becomes advantageous to compress a schedule, for example, by defining a maximum or an average cost of all time slots, above which compression should be performed. During scheduling, the heuristic can also be used as an optimization function.

IV. Related Work

The authors of [3] propose a compression algorithm that functions as a post-processing technique for scheduling, designed to consolidate multiple discrete time slots into a single time slot, thus diminishing the number of GCL entries and guardbands, which consequently yields an increase in the bandwidth allocated to low-priority traffic. Our heuristic also focuses on grouping time slots, but additionally considers the starvation of low-priority traffic and the requirements of reconfiguring running streams.

A path-based metric of flexibility is proposed in [4], denoted as *flexcurve*, which evaluates the feasibility of integrating new streams into the network by evaluating bottlenecks and considering frame size. This metric also accommodates the consideration of deadline requirements for streams. Furthermore, the structure of *flexcurve* enables simultaneous admission checks for multiple streams, thereby evaluating the network's capacity to accommodate additional streams. Like *flexcurve*, we also use a measure of flexibility for evaluating streams, however, our heuristic is used to evaluate the reconfiguration of existing streams without path alteration.

V. Conclusions

In this paper, we present a heuristic for the compressing of TSN schedules. This heuristic can be utilized to evaluate time slots and schedules according to various requirements, such as duration and number of time slot units, intervals between time slot units, reconfigurability of streams, link utilization, and minimization of stream latency and guard bands. The proposed heuristic can be used as an optimization function in scheduling, for post-processing of schedules, or for triggering and performing separate schedule compression. Our approach enables fine-grained adaptation of the heuristic through weighting and adjustment of the evaluation function, tailored to the specific requirements of the schedule. In our future work, we are going to integrate our heuristic approach into algorithms for schedule compression and online scheduling. Furthermore, we conduct a comprehensive analysis of various weighting schemes and evaluation function settings for different use cases.

Acknowledgments

This work has been achieved in the German 6GHealth project and has been funded by the German Federal Ministry of Education and Research under reference number 16KISK227.

References

[1] "IEEE standard for local and metropolitan area networks – bridges and bridged networks," *IEEE Std 802.1Q-2022 (Revision of IEEE Std 802.1Q-2018)*, 2022, doi: 10.1109/IEEESTD.2022.10004498.

[2] F. Kummer, F. Golatowski, W. Brekenfelder, H. Parzyjegla, P. Danielis, and G. Mühl, "An online scheduler for reconfigurable time-sensitive networks," in *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2024, pp. 01–08, doi: 10.1109/ETFA61755.2024.10711137.

[3] F. Dürr and N. G. Nayak, "No-wait packet scheduling for ieee time-sensitive networks (tsn)," in *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, ser. RTNS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 203–212. [Online]. Available: https://doi.org/10.1145/2997465.2997494

[4] C. Gärtner, "Flexibility in real-time networks : Analytical approaches for adapting time-sensitive networks to dynamic traffic requirements," Ph.D. dissertation, Technische Universität Darmstadt, Darmstadt, August 2024. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/27880/