Improving Latency Analysis for Flexible Window-Based GCL Scheduling in TSN Networks by Integration of Consecutive Nodes Offsets

Zhao, Luxi; Pop, Paul; Gong, Zijie; Fang, Bingwu

Published in:
IEEE Internet of Things Journal

Link to article, DOI:
10.1109/JIOT.2020.3031932

Publication date:
2021

Document Version
Peer reviewed version

Citation (APA):
Zhao, L., Pop, P., Gong, Z., & Fang, B. (2021). Improving Latency Analysis for Flexible Window-Based GCL Scheduling in TSN Networks by Integration of Consecutive Nodes Offsets. *IEEE Internet of Things Journal*, *8*(7), 5574 - 5584. https://doi.org/10.1109/JIOT.2020.3031932

Improving Latency Analysis for Flexible Window-Based GCL Scheduling in TSN Networks by Integration of Consecutive Nodes Offsets

Luxi Zhao, Paul Pop, Zijie Gong, and Bingwu Fang

Abstract—**Time-Sensitive Networking (TSN) is an upcoming set of Ethernet standards designed for real-time and safety-critical Internet of Things (IoT) applications in automotive, aerospace and industrial automation domains. With the combination, complexity and flexibility of flow control mechanisms in TSN connected systems, the performance analysis for mixed-critical messages is becoming a difficult challenge. The flexible window-based Gate Control List (GCL) scheduling model has been proposed as a relaxation to assumptions on frames-to-window allocation, mutually exclusive gates opening, and scheduled end systems and switches, which offers more flexibility in the configuration of GCLs. In this paper, we are interested in providing a reliable verification method based on the network calculus theory to drive GCL configurations for TSN networks. To the best of our knowledge, this is the first performance analysis method suitable for the general flexible window-based GCLs in entire TSN networks, by reflecting the relative positional relationships of windows for same priority queues on consecutive nodes and constructing the window limitations into the shaper curve, in order to reduce the pessimism of the latency bounds. We validate the proposed method through Industrial IoT synthetics test cases and two large realistic cases, showing the significant reduction in pessimism on delay bounds, and the correctness and scalability by comparing with results from the previous work and simulation results.**

I. Introduction

THE emergence of the Internet of Things (IoT) is driving the demand for higher levels of safety and reliability for cyber-physical systems. Time-Sensitive Networking (TSN) [1] is an enhancement of the traditional Ethernet and plays an increasingly significant role in safety-critical and real-time communications. TSN framework is promising to provide a real-time platform for IoT development. With the TSN amendments, multiple traffic classes from real-time to best-effort share the same network. A set of scheduling mechanisms [2] from TSN substandards have been proposed and can interact in several combinations.

In this paper, we focus on one set for safety-critical communication configured via Scheduled Traffic (ST, also called Time-Triggered traffic) [1, § 8.6.8.4] (previously defined in 802.1Qbv [3]) based on the timed gates at the output ports controlled by Gate Control Lists (GCLs). GCLs rely on a global synchronized clock (802.1ASrev [4]) available to devices engaging in the communication.

Several ways of using and configuring ST have been proposed [5]–[7], from which GCLs can be treated either to enable a more deterministic behavior of transmission for individual ST flows. Craciunas et al. [5] propose a method for GCLs configuration to enable zero jitter as well as fully-deterministic end-to-end latency for individual ST flows. The deterministic communication behavior is achieved by enforcing a complete isolation of critical flows from each other in the time or space domain. The disadvantages of this approach are that it limits the GCL synthesis solution space and takes a long time to solve the GCL synthesis in the case of large scale networks, which is an intractable problem [6]. In [7], ST frames are allowed to delay each other if they are in the same queue, benefiting from relaxing the strictly periodic constraints. However, all the above works necessitate that the traffic leaves the sending end systems in a scheduled and synchronized fashion (i.e., requiring TSN capabilities on both end systems and switches) and enforce mutually exclusive gate openings for multiple ST queues, since the schedule controls the transmission and forwarding of frames from sending node to receiving node within the network, requiring that the interference between ST frames is either 0 or bounded by the schedule construction. This is the main limitation of previous methods that they require end systems to be scheduled and synchronized to the network, i.e. to have 802.1Qbv and 802.1ASrev TSN capabilities. Nevertheless, this is often not the case since many systems use e.g. off-the-shelf sensor nodes without TSN mechanisms. In [8], [9], a class-based GCL was proposed without per-flow scheduling nor requirement of synchronized end systems. However, their scheduling models are quite limited as the offset of windows on different nodes is not considered assuming windows to be aligned among all switches. In this paper, we consider a more general flexible window-based scheduling model, i.e., besides the above constraint relaxation, windows do not have to be aligned and can be placed at any time slot on nodes in networks. Such a scheduling model supports both periodic and sporadic critical traffic, and better supports the reconfiguration of the networks.

Since the transmission of ST flows is no longer completely

Zhao was with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark, She is now with the Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany

Paul Pop is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark, 2800 DK e-mail: luxzha@dtu.dk; paupo@dtu.dk.

Zijie Gong and and Bingwu Fang are with the Department of Electronics and Information Engineering, Beihang University, Beijing, China, CN 100191 e-mail: gongzijie@buaa.edu.cn; bingwufang@buaa.edu.cn.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031932, IEEE Internet of Things Journal

deterministic due to the relaxed assumptions, it requires formal performance analysis methods to provide safe latency bounds. The method proposed in [8] for the worst-case delay bounds is not based on any formal verification methods. It is pessimistic for small scale networks, and is not "safe" for large scale networks. Network Calculus (NC) [10], [11] is a mature theory to calculate safe upper bounds for latencies and queuing backlogs of real-time traffic in communication networks. There are several studies addressing the performance analysis for window-based scheduling using network calculus approach. Timing analysis approaches [12], [13] have been proposed to guarantee the latency bounds for Time-Division Multiple Access (TDMA) bus protocol. However, they cannot be directly applied for the flexible window-based GCL model in TSN networks, since overlapping of time slots is not allowed in TDMA and due to the bus characteristics, there is no need to do performance analysis on consecutive nodes thus the timing analysis on such a TDMA bus is relatively easier. Zhao et al. [14] provided the latency analysis for a mixed ST traffic and priority-based scheduling when windows overlapping for different priority queues on an output port of a node happen. Nevertheless, the analysis is only suitable for a single node but not for the whole TSN switched network, as it did not consider the offsets, i.e., relative positional relationships, of windows for same priority queues on output ports of consecutive nodes. Thus it cannot be directly used for our more general flexible window-based scheduling model proposed in this paper. Using it, will bring a very large pessimism when calculating the upper bound of worst-case end-to-end delay (WCD) for the whole network.

The main contribution of this paper is to propose the timing analysis model for the general flexible window-based GCL scheduling to the whole network. We construct the service curve of NC for flexible window-based GCL, by considering not only the window length and period, but also to the relative positional relationship of windows from preceding nodes, which was not considered in [14]. For the overlapping situations of different priority windows, we refer the reader to the discussion in [14]. Taking the window offsets into account is not trivial as the window cycles could be different on consecutive nodes, and there may be multiple preceding nodes connected to the present node, which will be shown in detail in Sect. IV. In addition, the paper also contributes on limiting the arrival burst of ST flows from preceding nodes by constructing the window limitations into the shaper curve of NC, thus to reduce the pessimism of the analysis. Lastly, the paper evaluates the proposed method through several synthetic test cases and two large realistic cases. By comparing with the results calculated by directly using the previous work [14] and simulation respectively, we show the significant reduction in pessimism on delay bounds, and the correctness and the scalability of the proposed analysis. We also show the sensitivity to the window offsets on successive nodes, which means that the method proposed in this paper can help with the configuration design for ST traffic under the flexible window-based scheduling in TSN networks.

The remainder of the paper is organized as follows. Section II describes the network and traffic models. The network calculus theory is briefly introduced in Section III. Section IV describes the worst-case latency analysis for ST traffic transmitting in the whole network. Finally, the performance evaluation and the conclusion are given respectively in Sections V and VI.

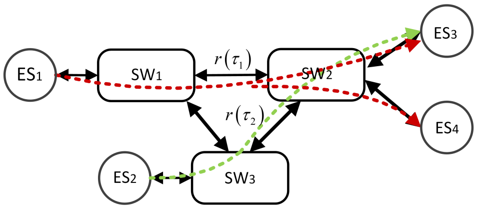
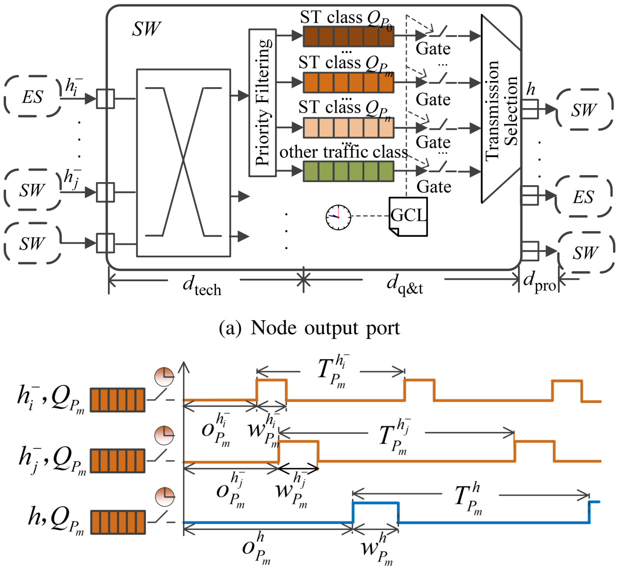


Fig. 1. TSN network topology.



(a) Node output port

(b) Windows for the same priority queues on different nodes

Fig. 2. TSN network architecture.

II. Network and ST Traffic Model

The TSN network model captures the end systems (ES), switches (SW) and physical links. In the following, "node" is used to represent an ES or a SW, and "node output port" is used to represent an output port of a node. The network topology is modeled as an undirected graph $G(\boldsymbol{E}, \boldsymbol{V})$, where $\boldsymbol{V} = \boldsymbol{ES} \bigcup \boldsymbol{SW}$ is the set of end systems and switches, and $\boldsymbol{E}$ is the set of physical links, which are full duplex, see Fig. 1 for an example. Without loss of generality we assume that all physical links and output ports in nodes have the same transmission rate C.

The scheduling mechanism of a SW in this paper is flexible window-based scheduling. An ES supports either strict priority scheduling or flexible window-based scheduling, which is decided by the system engineer. The architecture of an output port of a SW is shown in Fig. 2(a). Incoming flows are filtered through a switching fabric in the SW to the corresponding output ports according to the routes configured in the SW. Frames are enqueued in the associated priority queues according to the priority (traffic class) of the flow. There are 8 queues per output port, each of which is controlled by a gate with two states, open and closed. A set of queues Q_{P_m} are for ST traffic with priority P_m ($m \in [0, n]$, $n \in [0, 7]$), and the remaining queues are for other traffic classes, of which gate states are mutually exclusive from ST gates. Only when the gate is open, frames waiting in the queue are eligible to be forwarded, in first-in first-out (FIFO) order. IEEE 802.1Qbv [3] substandard proposes a lookahead mechanism for each ST traffic class to check if there is enough time to send the entire frame before the gate closed, in order to prevent frames from being transmitted after the corresponding gate closes. If there is not enough time, the frame cannot be forwarded until the next window, and there will be an idle time i.e., guard band (GB) in this paper, at the end of the window. Moreover, we consider only the case where the windows of different priority queues do not overlap with each other. For the case of multiple gates open at the same time, we refer readers to [14] and we will not discuss this further; the model proposed in this paper can be easily extended to the overlapping situation.

For each ST priority queue, there is a periodic window with a certain length w_{P_m}, period T_{P_m}, and offset o_{P_m} relative to a starting point, as shown in Fig. 2(b) for example. The window lengths and periods could be different for the same priority queues on different nodes, which depend on the GCLs configured based on the experience of the engineer or determined using tool such as [15], for example, Q_{P_m} on h, h_1^- and h_2^- in Fig. 2(b), where the nodes h_1^- and h_2^- connect to the node h in parallel.

The time-sensitive applications are modeled as messages which are sent from ESes via a set of ST flows τ_{ST}. For a ST flow $\tau_i \in \tau_{ST}$, we know its frame size l_i, the period (periodic flow) or the minimum time interval between two consecutive frames (sporadic flow) p_i before entering the source ES, and its priority. The set of ST flows with the same priority P_m is denoted as τ_{P_m}. The flow route $r(\tau_i)$ is statically defined, including multicast situations, see for example in Fig. 1. Table I summarizes the notations used in this paper.

III. Network Calculus Background

The timing analysis method in this paper is based on the Network Calculus [10], which is a well-established theory used to calculate upper bounds of latency and backlog for the certification of real-time networks. The bounds depends on the construction of arrival and service curves for the investigated flows and network nodes.

Definition 1: (Arrival Curve) Given a flow with the input cumulative function $R(t)$, which is the total number of bits of the flow that has arrived up to time t, $\alpha(t)$ is an arrival curve for $R(t)$ iff

$$R(t) \leq \inf_{0 \leq s \leq t} \{R(s) + \alpha(t-s)\} = (R \otimes^{(1)} \alpha)(t). \qquad (1)$$

Definition 2: (Strict Service Curve) If a node provides the service for a flow with the input cumulative function $R(t)$ and output cumulative function $R^*(t)$, then the node offers the strict service curve $\beta(t)$ iff during any backlogged period $(s,t]$[2]

$$R^*(t) - R^*(s) \geq \beta(t-s). \qquad (2)$$

TABLE I
Summary of notation.

Symbol	Meaning
ES	End systems
SW	Network switches
C	Physical link rate
P_m	Priority of the ST traffic
Q_{P_m}	Queue for ST traffic with priority P_m
w_{P_m}	Window length of GCL for queue Q_{P_m}
T_{P_m}	Window period of GCL for queue Q_{P_m}
o_{P_m}	Window offset of GCL for queue Q_{P_m}
τ_{ST}	Set of ST flows
τ_{P_m}	Set of ST flows with the priority P_m
τ_i	A ST flow $\in \tau_{ST}$
l_i	Frame size of τ_i
p_i	Period/minimum interval between consecutive frames of τ_i before ES
$r(\tau_i)$	Flow route of τ_i
h	Output port of a node
h^-	Output port of a preceding node connected to h
fWN,nfWN	First and non-first node of flexible window-based scheduling
$R_i^h(t)$,$R_i^*(t)$	Input and output cumulative functions of flow τ_i in h
$\alpha_i^h(t)$,$\alpha'^h_i(t)$	Input and output arrival curves of flow τ_i in h
$\alpha_{P_m}^{h,h^-}$	Input arrival curve of grouped aggregate ST flows with priority P_m in h from h^-
$\alpha_{P_m}^{h,h^-,t_*^h}$	A possible input arrival curve of aggregate ST flows with priority P_m in h from h^- based on t_*^h
$\alpha_{P_m}^{h,t_*^h}$	A possible input arrival curve of aggregate ST flows with priority P_m in h based on t_*^h
t_E^{h,h^-},t_L^{h,h^-}	Earliest and latest possible arrival times in h of frames obtained service within a window in h^-
$l_{P_m min}^h$	Minimum frame size with priority P_m in h
$l_{P_m max}^h$	Maximum frame size with priority P_m in h
t_*^h	A possible earliest starting time of backlogged period
$WT_{P_m}^{h,t_*^h}$	Maximum waiting time at the beginning of a backlogged period of ST flows with priority P_m in h relative to t_*^h
$HP_{P_m}^h$	Hyperperiod of windows for Q_{P_m} in h and h^-
$N_{P_m}^h$	Number of benchmark windows for Q_{P_m} in h
$\overline{w}_{P_m}^h$	Guaranteed (minimum) service of the window $w_{P_m}^h$
$\overline{W}_{P_m}^h$	Maximum service of the window $w_{P_m}^h$
$\beta_{P_m}^{h,t_*^h}(t)$	A possible service curve for aggregate ST flows with priority P_m in h based on t_*^h
$o_{P_m}^{h^-,t_*^h}$	Relative offset between t_*^h and t_E^{h,h^-}
$\delta_{D_q}(t)$	Burst-delay function
D_{tech}	Technical latency bound in SW
$D_{q,i}^h$	Queuing delay bound of flow τ_i in h
$D_{P_m}^{h,t_*^h}$	A possible delay bound in h based on t_*^h for ST traffic with priority P_m
$D_{P_m}^{h,h^-,t_*^h}$	A possible delay bound in h based on t_*^h for ST traffic from h^- with priority P_m
D_i^h	Queuing and transmission delay bound of flow τ_i in h

Definition 3: (Shaper Curve [11]) is a notation used to characterize the maximum number of bits that are served during any period $(s,t]$. $\sigma(t)$ is a shaper curve iff it is an arrival curve for all output $R^*(t)$, i.e.,

$$R^*(t) - R^*(s) \leq \sigma(t-s), \qquad (3)$$

which is equivalent with $R^*(t) \leq (R^* \otimes \sigma)(t)$. It is a concept of shaping the output cumulative function that will be guaranteed to be constrained by an arrival curve.

The delay bound D experienced by the flow in the node is the maximum horizontal distance between $\alpha(t)$ and $\beta(t)$ denoted as $h(\alpha, \beta)$. The network worst-case end-to-end latency of a

[1] $(f \otimes g)(t) = \inf_{0 \leq s \leq t}\{f(t-s) + g(s)\}$.
[2] $(s,t]$ is backlogged period if $R(t') - R^*(t') > 0$, $\forall t' \in (s,t]$.

flow is bounded by the sum of delay bounds in each network nodes along its route.

Definition 4: (Output Arrival Curve) $\alpha'(t)$ for a flow $R(t)$ of arrival curve $\alpha(t)$ crossing a server with the service curve $\beta(t)$ can by computed based on the queuing delay bound D_q waiting in the node, i.e.,

$$\alpha'(t) = (\alpha \oslash {}^{(3)}\delta_{D_q})(t). \tag{4}$$

where the queuing delay bound D_q is calculated by the delay bound D experienced by the flow in the node minus its transmission delay, and $\delta_{D_q}(t)$ is the burst-delay function [10] which is 0 if $t \leq D_q$ and ∞ otherwise. It is also the input arrival curve for the next server.

IV. Worst-Case Analysis for ST Traffic

Considering a ST flow, its sources of end-to-end delay are as follows (see Fig. 2(a)), along its route from the source ES through the SWs to the destination ESes: (i) The technical latency d_{tech} in the SW, which is from the time after the frame being fully received, to the time when it arriving at the queue after the switching fabric. The maximum value is known and denoted as D_{tech}. (ii) The queuing and transmission delay in the output ports of each node (both source ESes and SWs). It is upper bounded by D^h ($h \in \{ES, SW\}$), which will be discussed in Sect. IV-D. (iii) The propagation delay d_{pro} which is tightly related to the physical medium. We ignore it in this paper as it is negligible compared with other delays. It can easily be added as a constant value to the model as needed. Then the worst-case end-to-end delay (WCD) of a ST flow τ_i can be bounded by the sum of latency bounds in each network nodes along its route,

$$D_i = D_i^{ES} + \sum_{SW \in r(\tau_i)} \left(D_i^{SW} + D_{tech} \right). \tag{5}$$

As described in Sect. II, the scheduling type of SW is flexible window-based scheduling, i.e. no flow to window assignment. Then, for the first node (fWN) of flexible window-based scheduling (either the first SW if ES supports strict priority scheduling or the source ES if ES also supports flexible window-based scheduling), as there are no window constraints from the preceding node, the arrival time of flow is arbitrary. Thus, the worst-case timing analysis for ST flows on such a node fWN is same as with the TDMA bus [12], [13]. However, for the non-first node (nfWN) of flexible window-based scheduling, to obtain the upper bound latency of a ST flow τ_i on its node output port h, we need to consider the effect of relative positional relationships of windows of the same priority queues from preceding node output ports h^- connected to h. Note that in the following, we will only discuss the performance analysis for the node based on the flexible window-based scheduling. The timing analysis for the node with the strict priority scheduling can be found in [16].

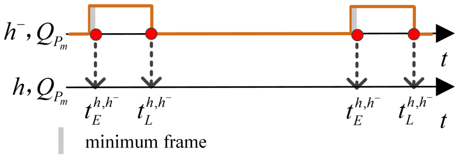


Fig. 3. Window limitations - the earliest and latest arrival times.

A. Impact of Windows on Traffic Arrival

On the one hand, as there is no flow to window assignment, the arrival time of frames from the preceding node h^- is not deterministic. On the other hand, since only when the gate is open, frames waiting in the queue of h^- are eligible for transmission. Thus, the arrival time of a frame on the non-first node nfWN=h based on flexible window-based scheduling is not arbitrary, and will be constrained by the window (i.e., the duration when the gate is open) from the previous node h^-.

See for example in Fig. 3, the gate for Q_{P_m} on a preceding node h^- is periodically open and close. Thus, the arrival times of frames passing from the node h^- to the node h are limited within $[t_E^{h,h^-}, t_L^{h,h^-}]$, where t_E^{h,h^-} and t_L^{h,h^-} are respectively the earliest and the latest possible arrival times on h of frames obtained service within some window on h^-. Considering the store-and-forward transmission, the reception time of the last bit of the frame is taken as the arrival time of the frame. Then the earliest arrival time t_E^{h,h^-} equals to the open time for the window on h^- plus the minimum frame transmission time $l_{P_m min}^{h^-}/C$, and the latest arrival time t_L^{h,h^-} equals to the latest sending time from h^-, i.e., the close time for the window on h^-.

The non-arbitrary arrival time of ST traffic from the preceding node h^- does not only affect on the service curve for ST traffic on the present node h, but also on the arrival curve of aggregate ST traffic to h, which will be be discussed respectively in Sect. IV-B and Sect. IV-C as follows.

B. Strict Service Curve for ST Traffic with Priority P_m

In this section, we discuss the service curve for ST traffic with priority P_m ($m \in [1, n]$, $n \in [1, 8]$) on a node output port with the consideration of window constraints from preceding nodes. Due to the shaping impact of windows from previous nodes on the arrival of frames in switched networks, the maximum waiting time at the beginning of a backlogged period is not only related to the window length and the period on the current node output port h, but also related to arrival limitations of frames from preceding node output ports h^-.

1) The maximum waiting time $WT_{P_m}^h$ at the beginning of a backlogged period: The maximum waiting time WT is defined as the maximum non-guaranteed service time of the frame at the beginning of a backlogged period. For the first node fWN based on the flexible window-based scheduling, since there are no window constraints from preceding nodes, the starting time of backlogged period on such node is arbitrary, which is same as for the TDMA bus [13]. Thus, the maximum waiting time WT happens when the first frame of a backlogged period has the maximum frame size and arrives at the instant t_* when the

[3] $(f \oslash g)(t) = \sup_{s \geq 0}\{f(t+s) - g(s)\}$.

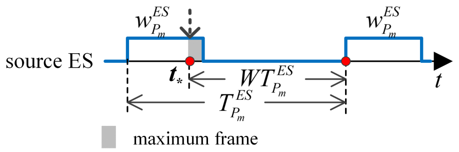
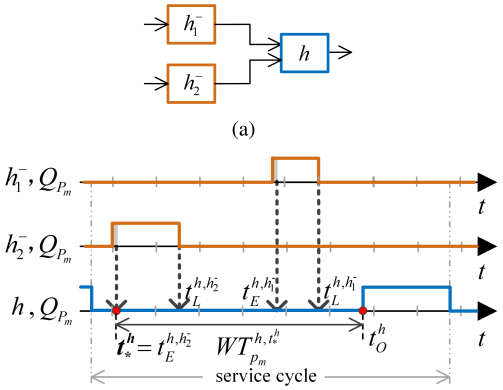


Fig. 4. WT at the beginning of the backlogged period for source ES.



(a)

(b)

Fig. 5. Relative positional relationships of windows in a service cycle.

remaining time during the present window is slightly less than the frame transmission time, as shown in Fig. 4, i.e.,

$$WT^{\text{fWN}}_{P_m} = l^{\text{fWN}}_{P_m max}/C + T^{\text{fWN}}_{P_m} - w^{\text{fWN}}_{P_m}, \tag{6}$$

where $l^{\text{fWN}}_{P_m max}$ is the size of maximum frame with priority P_m sent through the output port of fWN, and $T^{\text{fWN}}_{P_m}$ and $w^{\text{fWN}}_{P_m}$ are respectively the window cycle and length for the queue Q_{P_m} on the fWN. Note that fWN represents the first SW after an ES if the ES does not support flexible window-based scheduling, and represents an ES if the ES supports flexible window-based scheduling.

However, for the non-first node nfWN=h based on flexible window-based scheduling, there are sending time constraints from preceding nodes, as discussed in Sect. IV-A. Thus, the start of the backlogged period cannot happen at anytime, but it is related to arrival limitations. For example, in Fig. 5(a), there are two preceding nodes h_1^- and h_2^- connected to the present node h. The relative positional relationships of windows of queue Q_{P_m} among h_1^-, h_2^- and h in one service cycle are shown in Fig. 5(b). The service cycle represents the window period $T^h_{P_m}$ of queue Q_{P_m} on the present node h to be studied. If taking the service window in such a service cycle on h as a reference (benchmark), it means that such window is the first service time slot for the latest backlogged period. Since the traffic arrival on h from preceding nodes h^- is limited within $\left[t_E^{h,h_1^-}, t_L^{h,h_1^-}\right]$ and $\left[t_E^{h,h_2^-}, t_L^{h,h_2^-}\right]$ in Fig. 5, we can find that when the start time t_*^h of backlogged period equals to t_E^{h,h_2^-}, there exists the maximum waiting time at the beginning of the backlogged period,

$$WT^{h,t_*^h}_{P_m} = t_O^h - t_*^h, \tag{7}$$

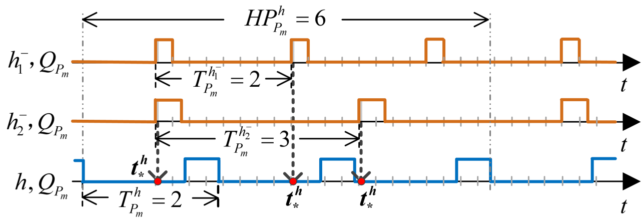


Fig. 6. Window cycles and the hyperperiod.

where t_*^h is the earliest possible starting time of backlogged period with consideration of arrival limitations from preceding nodes, which equals to t_E^{h,h_2^-} in Fig. 5 for example, t_O^h is the gate opening time of the service window benchmark for the corresponding priority queue on the node output port h.

Moreover, it is worth to note that the case of relative positional relationships of windows on preceding node output ports h^- and the present node output port h is not unique, as the window cycles for the same priority queues could be different on nodes h^- and h, see for example in Fig. 6 of two preceding nodes h_1^- and h_2^- and present node h with window period respectively of $T^{h_1^-}_{P_m} = T^h_{P_m} = 2$ and $T^{h_2^-}_{P_m} = 3$. However, all possible cases of relative positional relationships of windows on h^- and h are limited within the hyperperiod $HP^h_{P_m}$, which is defined as the least common multiple (LCM) of window cycles for the same priority queues on the present node output port h and all the preceding node output ports h^- directly connected to h, i.e.,

$$HP^h_{P_m} = LCM_{h_n^- \in [h^-,h]}\left(T^{h_n^-}_{P_m}, T^h_{P_m}\right), \tag{8}$$

where $[h^-, h]$ represents physically connected h^- and h. For example, in Fig. 6, $HP^h_{P_m} = LCM\left(T^{h_1^-}_{P_m}, T^{h_2^-}_{P_m}, T^h_{P_m}\right) = LCM(2,3,3) = 6$. Therefore, there are at most

$$N^h_{P_m} = HP^h_{P_m}/T^h_{P_m} \tag{9}$$

possible window relationships, if choosing a window in the hyperperiod on h as the benchmark. Corresponding to each window benchmark, it is possible to find a start time t_*^h leading to the maximum waiting time of the backlogged period. For example in Fig. 6, there are three possible starting times of backlogged period by selecting different windows on h within the hyperperiod as benchmark.

2) The guaranteed service $\overline{w}^h_{P_m}$ of the window $w^h_{P_m}$: The lower bound service offered by each window depends on the maximum frame size passing through and the integration mode selection. Since as mentioned in Sect. II, there will be a guard band if the frame cannot finish its transmission before the gate close event. Meanwhile, the guaranteed service slot $\overline{w}^h_{P_m}$ is at least larger than the minimum frame transmission $l^h_{P_m min}/C$. Thus, $\overline{w}^h_{P_m}$ is given by,

$$\overline{w}^h_{P_m} = max\left\{w^h_{P_m} - l^h_{P_m max}/C, l^h_{P_m min}/C\right\}, \tag{10}$$

where $l^h_{P_m max}$ and $l^h_{P_m min}$ are respectively the maximum and minimum frame length.

3) A possible strict service curve $\beta^{h,t^h_*}_{P_m}(t)$ based on a window benchmark for ST traffic: As discussed in Sect. IV-B1, the service curve for ST traffic with priority P_m on h will be separately derived based on each window benchmark selection in the hyperperiod.

Theorem 1: The possible strict service curve for ST traffic with priority P_m on a node output port h based on a window benchmark selected is given by

$$\beta^{h,t^h_*}_{P_m}(t) = \beta_{T^h_{P_m},\overline{w}^h_{P_m}}\left(t + T^h_{P_m} - \overline{w}^h_{P_m} - WT^{h,t^h_*}_{P_m}\right), \qquad (11)$$

where $WT^{h,t^h_*}_{P_m}$ and $\overline{w}^h_{P_m}$ can be determined respectively by Eq. (7) and Eq. (10), and $\beta_{T,w}(t)$ is the classic service curve for TDMA bus protocol [12],

$$\beta_{T,w}(t) = C \cdot max\left(\left\lfloor \frac{t}{T} \right\rfloor \cdot w, t - \left\lceil \frac{t}{T} \right\rceil \cdot (T - w)\right). \qquad (12)$$

For the proof of Theorem 1, we refer readers to the proof for Theorem 1 in [14]. We have a similar proof strategy, but with the different consideration for the maximum waiting time $WT^{h,t^h_*}_{P_m}$ at the beginning of the backlogged period.

C. Arrival Curve for Aggregate ST Traffic with Priority P_m

1) Arrival curve for individual flows: Since in the source ES h_0, each ST flow τ_i is known with the frame size l_i and the period (periodic flow) or the minimum time interval between two consecutive frames (sporadic flow) p_i, the input arrival curve of ST flow τ_i in h_0 can be given by,

$$\alpha^{h_0}_i(t) = \left\lceil \frac{t}{p_i} \right\rceil \cdot l_i, \qquad (13)$$

if $t \geq 0$, and 0 otherwise. The staircase function is considered here to obtain a more fine granularity for the arrival curve model. The input arrival curve of ST flow τ_i in the present h, which is also the output arrival curve for τ_i from the preceding h^-, can be calculated from Def. 4,

$$\alpha^h_i(t) = \alpha^{h^-}_i(t) \oslash \delta_{D^{h^-}_{q,i}}(t), \qquad (14)$$

where $D^{h^-}_{q,i}$ is the worst-case queuing delay of ST flow τ_i waiting in the previous h^-. It is calculated from $D^{h^-}_i - l_i/C$, where $D^{h^-}_i$ is the worst-case latency of flow τ_i in the node output port h^-, which given by Eq. (22) and will be discussed in Sect. IV-D.

2) Arrival curve for aggregate flows with priority P_m: The aggregate ST flows of priority P_m arriving on a node output port h could be from multiple preceding node output ports h^-. If ST flows from the same preceding h^- are taken as a group, the input arrival curve for grouped aggregate ST flows before h is constrained by the three aspects: (i) the sum of output arrival curves of individual flows from the preceding h^-; (ii) the shaper curve $\sigma_{link}(t)$ of physical link rate, which means that grouped flows cannot arrive on h at the same time,

$$\sigma_{link}(t) = C \cdot t; \qquad (15)$$

(iii) the shaper curve $\sigma^{h^-}_{P_m}(t)$ of windows from a preceding h^-, which means that the arrival of grouped ST flows on h is shaped by windows of queue Q_{P_m} on h^-, and will be discussed in detail afterwards. Then, the input arrival curve of grouped aggregate ST flows with priority P_m from the preceding h^- can be given by,

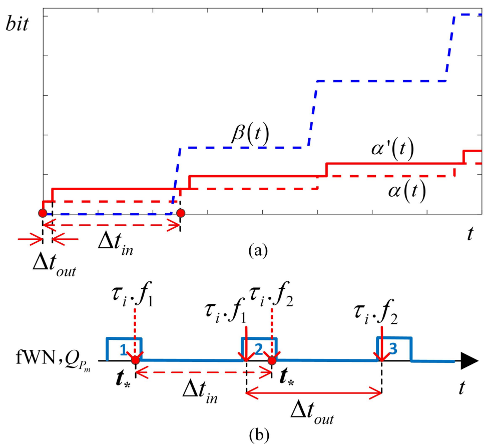


Fig. 7. Example of burst pessimism.

$$\alpha^{h,h^-}_{P_m}(t) = \left(\sum_{\substack{\tau_i \in \tau_{P_m} \\ h \in r(\tau_i)}} \alpha^h_i(t)\right) \bigwedge \left(\sigma_{link}(t) + l^{h,h^-}_{P_m max}\right) \bigwedge \sigma^{h^-}_{P_m}(t). \qquad (16)$$

Theorem 2: The shaper curve $\sigma^{h^-}_{P_m}(t)$ of windows for priority P_m queue from the preceding h^- is given by,

$$\sigma^{h^-}_{P_m}(t) = C \cdot min\left\{\left\lceil \frac{t}{T^{h^-}_{P_m}} \right\rceil \overline{W}^{h^-}_{P_m}, t - \left\lfloor \frac{t}{T^{h^-}_{P_m}} \right\rfloor \left(T^{h^-}_{P_m} - \overline{W}^{h^-}_{P_m}\right)\right\}, \qquad (17)$$

where

$$\overline{W}^{h^-}_{P_m} = \begin{cases} w^{h^-}_{P_m} & , h^- \neq \text{fWN} \\ \min\left\{w^{\text{fWN}}_{P_m}, \alpha^{\text{fWN}^-}_{P_m}\left(T^{\text{fWN}}_{P_m}\right)/C\right\} & , h^- = \text{fWN} \end{cases}$$

where $\alpha^{\text{fWN}^-}_{P_m}\left(T^{\text{fWN}}_{P_m}\right) = \lim_{t \to T^{\text{fWN}}_{P_m}, t < T^{\text{fWN}}_{P_m}} \alpha^{\text{fWN}}_{P_m}(t)$.

Proof: As known from the definition of shaper curve (Definition 3) in Sec. III, it is required to derive $R^{*h^-}_{P_m}(t + \Delta t) - R^{*h^-}_{P_m}(t) \leq \sigma^{h^-}_{P_m}(\Delta t)$ during any interval Δ to obtain $\sigma^{h^-}_{P_m}(\Delta t)$, where $R^{*h^-}_{P_m}(t)$ is the output cumulative function of flows with priority P_m on node output port h^-.

Since the window of length $w^{h^-}_{P_m}$ repeats with $T^{h^-}_{P_m}$, then in the best-case for ST flows with priority P_m, the maximum continuous access permission to service is $\overline{W}^{h^-}_{P_m} = w^{h^-}_{P_m}$. After such a duration, flows have no access to the link due to the gate close, and wait at least the interval of $T^{h^-}_{P_m} - \overline{W}^{h^-}_{P_m}$ till the gate opening again.

Especially, for the first node fWN of flexible window-based scheduling, the maximum waiting time at the beginning of the backlogged period is much larger relative to the window-based scheduling nodes afterwards, as discussed in Sect. IV-B1. Such a waiting time will lead to a large burst calculated by

Eq. (14) before the next flexible window-based scheduling node nfWN. However, in reality it cannot happen, as discussed in the following example. Let us assume that there is a periodic (resp. sporadic) flow $\tau_i \in \tau_{P_m}$ with period (resp. minimum interval between two consecutive frames) p_i and frame length l_i, which is served by the node fWN of flexible window-based scheduling with window period $T_{P_m} = p_i$ and window length $w_{P_m} \geq l_i$. The input arrival curve $\alpha(t)$ of τ_i and the service curve $\beta(t)$ supplied for it on the node fWN are respectively shown with red and blue dashed lines in Fig. 7(a). Then, the output arrival curve $\alpha'(t)$ for the flow calculated by Eq. (14) is represented by the red solid line. It can be seen from the figure that in the worst-case, two frames of τ_i become much closer ($\Delta t_{out} << \Delta t_{in}$) after being served by the node fWN. Nevertheless, this cannot happen in reality as shown in Fig. 7(b). The arrival time of frame $\tau_i.f_1$ is t_* on the window 1. If the service to it is delayed to the next window 2 because of the remaining time in window 1 is insufficient for the frame transmission, then the next frame instance will $\tau_i.f_2$ encounter the same situation. This is due to $T_{P_m} = p_i$, no matter what is the length of the window. Thus actually, $\Delta t_{out} = \Delta t_{in}$.

Therefore, if the window length is designed to consider the number of ST flow, such pessimism of the output burst from the first node fWN can be limited by the window length as discussed above. If the window length is designed larger than the real number of traversing flows for the reservation for the future new flows, the output burst calculated from Eq. (14) will be very pessimistic, and such pessimism cannot be alleviated by constructing the window into the shaper curve. Thus, we will additionally consider the maximum arrival bits of frames within the window period to limit the output arrival curve for the first node fWN of flexible window-based scheduling, which is given by $\alpha_{P_m}^{\text{fWN}^-}\left(T_{P_m}^{\text{fWN}}\right) = \lim_{t \to T_{P_m}^{\text{fWN}}, t < T_{P_m}^{\text{fWN}}} \alpha_{P_m}^{\text{fWN}}(t)$. Then for the first node fWN, the maximum service duration in a window is $\overline{W}_{P_m}^{\text{fWN}} = \min\left\{w_{P_m}^{\text{fWN}}, \alpha_{P_m}^{\text{fWN}^-}\left(T_{P_m}^{\text{fWN}}\right)/C\right\}$.

Therefore, in the best-case, the frames in the output queue $Q_{P_m}^{h^-}$ can obtain the service at most $C \cdot \Delta$ during any time interval $0 \leq \Delta t < \overline{W}_{P_m}^{h^-}$, but will not be offered any service during any time interval $\overline{W}_{P_m}^{h^-} \leq \Delta < T_{P_m}^{h^-}$. Then we can get,

$$R_{P_m}^{*h^-}(t + \Delta t) - R_{P_m}^{*h^-}(t) \leq$$
$$C \cdot min\left\{\left\lceil \frac{\Delta t}{T_{P_m}^{h^-}} \right\rceil \overline{W}_{P_m}^{h^-}, \Delta t - \left\lfloor \frac{\Delta t}{T_{P_m}^{h^-}} \right\rfloor \left(T_{P_m}^{h^-} - \overline{W}_{P_m}^{h^-}\right)\right\}.$$

■

Moreover, the relative positional relationships of windows on different preceding node output ports lead to the temporal separations of corresponding frame transmission. Thus, frames from different preceding nodes may not arrive at the same time on h. Then the arrival curve of aggregate ST flows on h will be introduced too much pessimism if we simply sum the arrival curves of grouped aggregate ST flows from all preceding node output ports. With the determination of the starting time t_*^h of the backlogged period based on a window benchmark on h, the relative positional relationship of windows is characterized by the relative offset $o_{P_m}^{h^-,t_*^h}$, which is defined as the interval between t_*^h and the earliest arrival time t_E^{h,h^-} arriving from h^- on h no earlier than t_*^h. For example in Fig. 8, relative offsets are given with different window benchmark selection. Note that $o_{P_m}^{h^-,t_*^h} = 0$ if $t_*^h = t_E^{h,h^-}$. Then, by taking into account the relative offset, the input arrival curve of grouped aggregate ST flows of priority P_m from h^- in Eq. (16) can be updated to,

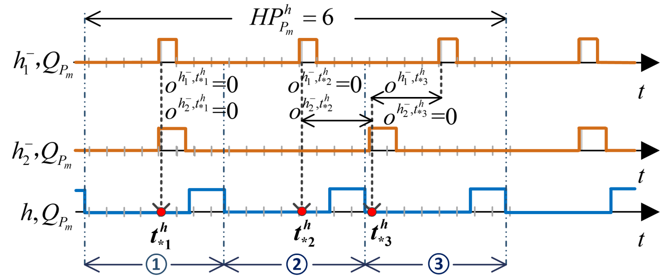


Fig. 8. Starting points t_*^h based on different window benchmarks; relative offsets based on t_*^h.

$$\alpha_{P_m}^{h,h^-,t_*^h}(t) = \alpha_{P_m}^{h,h^-}(t - o_{P_m}^{h^-,t_*^h}). \tag{18}$$

Thus, based on a window benchmark selected on h in the hyperperiod, the possible arrival curve for aggregate ST flows with priority P_m before the node output port h is given by,

$$\alpha_{P_m}^{h,t_*^h}(t) = \sum_{h^- \in r(\tau_{P_m})} \alpha_{P_m}^{h,h^-,t_*^h}(t). \tag{19}$$

D. Worst-case Latency for ST Traffic on a Node Output Port

According to the Network Calculus theory, one possible upper bound latency on the node output port h is the maximum horizontal deviation between the possible arrival curve $\alpha_{P_m}^{h,t_*^h}(t)$ in Eq. (19) and the corresponding possible service curve $\beta_{P_m}^{h,t_*^h}(t)$ in Eq. (11), obtained based on a selected t_*^h,

$$D_{P_m}^{h,t_*^h} = h\left(\alpha_{P_m}^{h,t_*^h}(t), \beta_{P_m}^{h,t_*^h}(t)\right). \tag{20}$$

Especially, with the determination of t_*^h, the earliest arrival time of frame from the preceding node output port h^- is always $o_{P_m}^{h^-,t_*^h}$ later than t_*^h, as can be seen from Fig. 5 for example. It also means that the start of the backlogged period for P_m traffic on h arriving from a preceding h^- is always $o_{P_m}^{h^-,t_*^h}$ late, and the worst-case delay for ST flows on node output port h should subtract the additional delay $o_{P_m}^{h^-,t_*^h}$, or the calculation will be excessive pessimism. Thus, we separately consider the worst-case delay on h of P_m traffic from different preceding node output ports h^-,

$$D_{P_m}^{h,h^-,t_*^h} = \max\left\{ D_{P_m}^{h,t_*^h} - o_{P_m}^{h^-,t_*^h}\Big|_{h^- \in r(\tau_{P_m})}, 0 \right\}. \tag{21}$$

Then the latency bound for the ST flow τ_i of priority P_m on the present h is the maximum value of all $N_{P_m}^h$ possible delay bounds based on a selected t_*^h,

$$D_i^h = \max_{t_*^h[N_{P_m}^h]} \left\{ D_{P_m}^{h,h^-,t_*^h} \right\}\Big|_{\substack{\tau_i \in \tau_{P_m} \\ h^- \in r(\tau_i)}}. \tag{22}$$

V. Experimental Evaluation

Two sets of experiment will be performed. One is conducted on several synthetic test cases, inspired from Industrial IoT use cases, to show the tightness improvement of the bounds from the proposed method compared to [14] and the correctness compared with simulation results. The other is run on two realistic use cases, i.e. an aerospace test case adapted from NASA's Orion Crew Exploration Vehicle (CEV) [17] and a test case related to interconnected vehicles (where vehicle-to-vehicle messages are disseminated within the in-vehicle TSN network), adapted from General Motors (GM) [18]. The realistic test cases evaluate the scalability of the analysis and the influence of relative positional relationships of windows from different consecutive node output ports on the latency upper bounds.

A. Synthetic Test Cases

In this section, to compare with the previous work [14], we use the synthetic test cases adapted from [14]. We name the method proposed in [14] as STNode, which means that it is suitable for single node analysis. However, since the STNode method does not consider the window offsets from consecutive node output ports, it will lead too much pessimism when analyzed for the whole network. Our new analysis method is named as STNet, as it fits for analyzing ST traffic in the whole TSN network. With the following experiment, we are interested to show the huge improvement of tightness of bounds from our new method compared to STNode. Meanwhile, to show the correctness of our method, we do the simulation analysis (named Sim) using NeSTiNg [19]. NeSTiNg is the extension of the OMNeT++/INET framework for the TSN network simulations.

The topology consists 2 SWes in series connection, each connected to 3 ESes via physical links with a rate of 1 Gb/s. We have 13 ST flows of different priorities from 0 to 6 with routes statically defined. The GCLs' window length, period and offset for each priority on each node output port are known. Like [14], we investigate the WCD of ST flow τ_{ST_1} to see the effect of changes in GCL. τ_{ST_1} is initially configured of the priority of 1 along the route $[[ES_2, SW_1], [SW_1, SW_2], [SW_2, ES_6]]$. There are five subsets of experiments adapted from [14], with additional consideration of parameters of relative positional relationships of windows along consecutive nodes. Each subset includes several scenarios of changing related parameters of GCLs along the route of τ_{ST_1}.

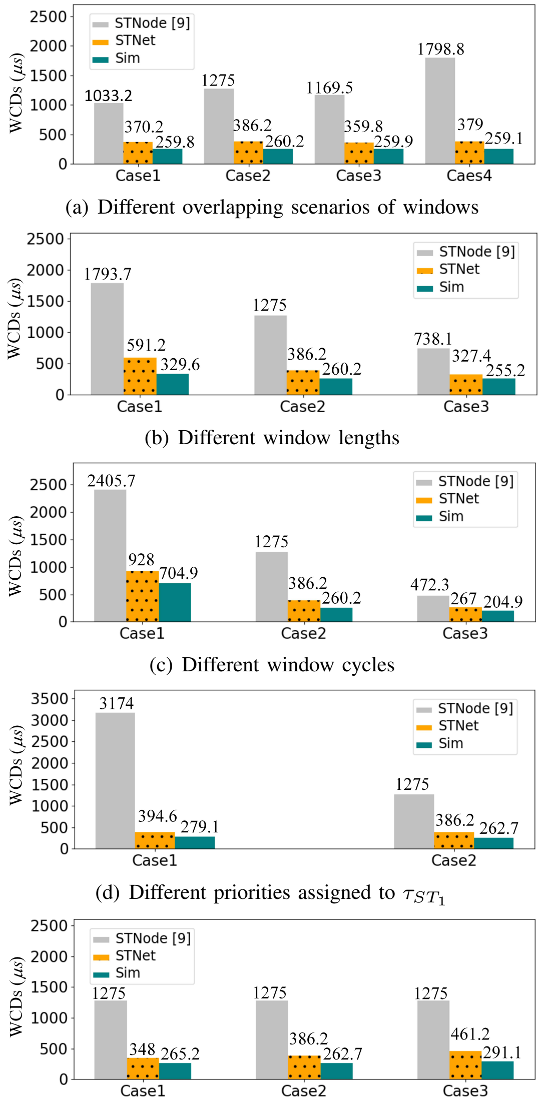
In the first subset of experiment, we evaluate how the overlapping variation of different priority windows on the same node output port affects the WCD of τ_{ST_1}. There are four test scenarios by changing overlapping situations for the priority queue along the route of τ_{ST_1}. For the other two experiments, we want to determine the influence on the WCD bound of τ_{ST_1}, respectively, of the length changing of the window and the variations in window cycle. The fourth one is to show the impact of different priorities allocated to τ_{ST_1} on its WCD. For the last synthetic test case, we keep the relative positional relationships of windows for all the priority queues along consecutive nodes unchanged except for the priority queues for τ_{ST_1}. The detailed descriptions of GCLs for τ_{ST_1} along its route under scenarios for each experiment are given in Table II. Case 2 is taken as the benchmark, which means that it gives the initial configuration of GCL and configurations for all the other scenarios of subsets are compared with it. As relative positional relationships of windows along consecutive nodes are not considered in STNode method, for fair comparison, synthetic test cases in [14] are adjusted by taking window offsets into account. The window offsets on source ES of τ_{ST_1} are assumed as 0.

TABLE II
GCLs along the route of τ_{ST_1}

Case	Link	Open-Close (μs)	Offset (μs)	Period (μs)	Priority
Benchmark test case					
2	$[ES_2, SW_1]$	[95, 115]	0	250	1
	$[SW_1, SW_2]$	[155, 175]	60		
	$[SW_2, ES_6]$	[180, 200]	85		
(a) Different overlapping scenarios of windows					
1	$[ES_2, SW_1]$	[105, 125]	0	250	1
	$[SW_1, SW_2]$	[165, 185]	60		
	$[SW_2, ES_6]$	[190, 210]	85		
3	$[ES_2, SW_1]$	[80, 100]	0	250	1
	$[SW_1, SW_2]$	[140, 160]	60		
	$[SW_2, ES_6]$	[165, 185]	85		
4	$[ES_2, SW_1]$	[65, 85]	0	250	1
	$[SW_1, SW_2]$	[125, 145]	60		
	$[SW_2, ES_6]$	[150, 170]	85		
(b) Different window lengths					
1	$[ES_2, SW_1]$	[95, 110]	0	250	1
	$[SW_1, SW_2]$	[155, 170]	60		
	$[SW_2, ES_6]$	[180, 195]	85		
3	$[ES_2, SW_1]$	[95, 125]	0	250	1
	$[SW_1, SW_2]$	[155, 185]	60		
	$[SW_2, ES_6]$	[180, 210]	85		
(c) Different window cycles					
1	$[ES_2, SW_1]$ $[SW_1, SW_2]$ $[SW_2, ES_6]$	[95, 115] [155, 175] [180, 200]	0 60 85	350	1
3				100	1
(d) Different priorities assigned to $\tau_{\mathbf{ST_1}}$					
1	$[ES_2, SW_1]$ $[SW_1, SW_2]$ $[SW_2, ES_6]$	[95, 115] [155, 175] [180, 200]	0 60 85	250	4
(e) Different window offsets on consecutive nodes					
1	$[ES_2, SW_1]$	[145, 165]	0	250	1
	$[SW_1, SW_2]$	[165, 185]	20		
	$[SW_2, ES_6]$	[190, 210]	45		
3	$[ES_2, SW_1]$	[95, 115]	0	250	1
	$[SW_1, SW_2]$	[180, 200]	85		
	$[SW_2, ES_6]$	[230, 250]	135		

The comparison results of five subsets of experiments are shown in Fig. 9. As we can see from the figure, our method reflects the same trend of latency bounds for τ_{ST_1} as the STNode method, but it significantly reduces the pessimism of the latency bounds compared to STNode. For these synthetic test cases, STNet is able to reduce the WCD bounds with 63.2% on average and 72.7% in maximum. Moreover, the simulation experiments are repeated 500 times randomly generating arrival time of ST flows in the source ES under each scenario. The maximum simulation results for τ_{ST_1} under different scenarios are shown in Fig. 9. In general, simulation results are usually more "optimistic" than the exact worst-case delay since simulation cannot guarantee exposing all possible

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031932, IEEE Internet of Things Journal



(a) Different overlapping scenarios of windows

(b) Different window lengths

(c) Different window cycles

(d) Different priorities assigned to τ_{ST_1}

(e) Different window offsets on consecutive nodes

Fig. 9. WCD of τ_{ST_1} with different scenarios.

scenarios. One the one hand, it shows the correctness of our improved model as the delay upper bounds obtained by STNet are always larger than the maximum delays obtained by Sim. On the other hand, for the flow τ_{ST_1}, the simulation results are 30% in average and 44% at most smaller than the results obtained by our approach STNet. It shows to some extent the upper bound of pessimism of STNet approach, which comes from the phenomenon of “pay burst only once” [10] in the basic Network Calculus approach. It is used to obtain an optimized global service curve of consecutive nodes of a flow, instead of summing up worst-case delays in each node along its route. However, such theory cannot be used in our model, as flows may enter and leave on any node in the network.

B. Realistic Test Case

In the last set of experiments, we use two real-life test cases adapted from the Orion CEV [20] and General Motors GM to show scalability of the analysis and evaluate the influence of relative positional relationships of windows along consecutive nodes on the WCD bounds of ST flows. For the Orion CEV case, it is assumed that all nodes in the network supports TSN capabilities, while for the GM case, it is assumed that only SW in the network support TSN mechanisms and ES supports only strict priority (SP) scheduling. The topology of Orion CEV in our paper is same with [20], including 31 ESes, 15 SWes, 100 dataflow routes connected by physical links transmitting at 1 Gb/s, and 100 ST flows with priority 0 to 7. GM has the topology of 20 ESes and 20 SWes, connected via physical links transmitting at 100 Mb/s. There are 27 ST flows with two priorities 0 and 1 running on the network. The GCLs for each priority queue on different nodes are manually generated for each realistic test case.

TABLE III
AVERAGE RELATIVE OFFSETS BETWEEN WINDOWS ON ADJACENT NODES

(a) Orion CEV

	Average window offset (μs)							
Priority	0	1	2	3	4	5	6	7
Case 1	655	903	1152	1400	2392	3385	3385	6362
Case 2	165	165	165	165	165	165	165	165

(b) General Motors

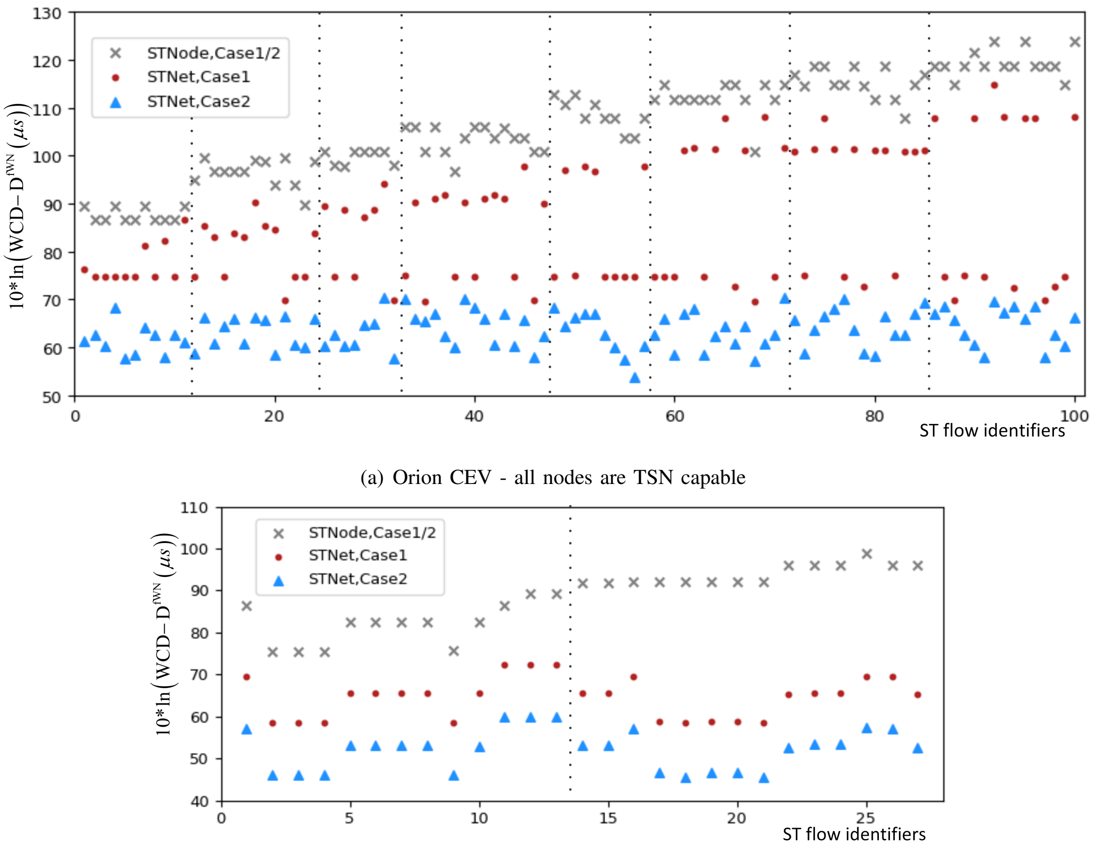
	Average window offset (μs)	
Priority	0	1
Case 1	403	403
Case 2	115	115

Each experiment has two subsets (Cases 1 & 2), which keep the window length and period situation of different priority queues on the same output port unchanged, but only change the window offsets of same priority queues on different nodes. Table III (a) and (b) respectively give the average relative offsets between windows for the same priority queues of all two connected nodes in networks for Orion CEV and GM. For each case, we use both STNode and STNet methods to calculate the latency upper bounds for ST flows.

As latency bounds calculation in the source ES (resp. the first SW) for the Orion CEV (resp. GM) is similar to the traditional TDMA bus, and the latency upper bounds in such a node in each subset are the same, in order to clearly show the paper contribution that the effect of relative positional relationships of windows on the latency bounds in the switched network, the experimental results shown in Fig. 10 are the WCD bounds after the first node fWN based on the flexible window-based scheduling. The y-axis in Fig. 10 uses a logarithmic scale with $10 \times ln(WCD - D^{fWN})$ and the x-axis represents the identifiers of ST flows. ST flows with different priorities are separated with vertical dotted lines in Fig. 10. As expected, the results in both Cases 1 & 2 calculated from STNode method are the same and extremely large, as shown with gray “x” dots. This is because that STNode does not take into account the relative locations of windows on consecutive nodes, but supposes that a frame arriving from the previous node has to wait longer at the beginning of the backlogged period on the present node. In other words, STNode method is not sensitive to the window offsets on successive nodes, hence is only suitable for a single node analysis.

Moreover, Fig. 10 also shows the results calculated by STNet

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.3031932, IEEE Internet of Things Journal



(a) Orion CEV - all nodes are TSN capable

(b) GM - only SWes are TSN capable

Fig. 10. Compared WCD bounds in STNode and STNet methods by varying of window offsets.

method, as given by red dots for Case 1 and blue triangles for Case 2. It shows very well that WCD upper bounds of ST flows decrease as the window offsets are decreasing. The results indicate that the method proposed in this paper can help with the configuration design for ST traffic under the flexible window-based scheduling in TSN networks. Meanwhile, STNet method is suitable for the timing analysis of ST traffic in the whole network, and significantly reduce the pessimism on estimating the network's latency bounds compared with STNode method.

VI. Conclusion

Formal performance analysis methods, such as network calculus, play an important role in the verification and validation of the correct real-time behavior of safety- and time-critical communication flows. This paper has presented a network calculus-based method to evaluate the flexible window-based GCLs, which covers the most generic use case for ST transmission in TSN networks. With the consideration of the relative positional relationships of windows for same priority queues on output ports of consecutive nodes and constructing the window limitations of preceding nodes into the shaper curve, the analysis method significantly reduces the pessimism compared to the previous work. Our proposed approach is the first work that can evaluate the entire TSN network under the flexible window-based GCLs. Because the method is sensitive to the relative positional relationships of windows in a GCL, it can also be used as a feedback to drive the synthesis flexible window-based GCL configurations.

Acknowledgment

The authors would like to thank to Anais Finzi from TT-Tech Computertechnik AG for the valuable discussion on the reducing pessimism of the analysis in the paper.

References

[1] IEEE, "802.1Q—IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks," https://standards.ieee.org/standard/802_1Q-2018.html, 2018.

[2] A. Nasrallah, A. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ULL) networks: A comprehensive survey covering the IEEE TSN standard and related ULL research," *arXiv preprint arXiv:1803.07673*, 2018.

[3] IEEE, "802.1Qbv—Enhancements for Scheduled Traffic," http://www.ieee802.org/1/pages/802.1bv.html, 2015.

[4] IEEE, "802.1ASrev—Timing and Synchronization for Time-Sensitive Applications," http://www.ieee802.org/1/pages/802.1AS-rev.html, 2017.

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

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

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

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

[9] N. Reusch, L. Zhao, S. S. Craciunas, and P. Pop, "Window-Based Schedule Synthesis for Industrial IEEE 802.1Qbv TSN Networks," in *Proc. of 16th IEEE International Conference on Factory Communication Systems*, 2020.

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

[11] A. Bouillard, M. Boyer, and E. Le Corronc, "Deterministic Network Calculus: From Theory to Practical Implementation," John Wiley & Sons, 2018.

[12] E. Wandeler, and L. Thiele, "Optimal TDMA time slot and cycle length allocation for hard real-time systems," in *Proceedings of IEEE Asia and South Pacific Design Automation Conference*, 2006.

[13] D. D. Khanh and A. Mifdaoui, "Timing analysis of TDMA-based networks using network calculus and integer linear programming." in *Proc. of IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*, 2014.

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

[15] N. Reusch, "Flexible Scheduling for the Time-Triggered Traffic in Time-Sensitive Networking", Masters thesis, Technical University of Denmark, 2019.

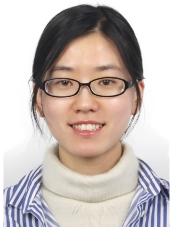
[16] J. Schmitt, P. Hurley, M. Hollick, and R. Steinmetz, "Per-flow guarantees under class-based priority queueing," in *IEEE Global Telecommunications Conference*, 2003.

[17] L. Zhao, P. Pop, Q. Li, J. Chen, and H. Xiong, "Timing analysis of rate constrained traffic in TTEthernet using network calculus," Real-Time System, vol. 52, no. 2, pp. 254-287, 2017.

[18] V. Gavrilut, B. Zarrin, S. Samii and P. Pop, "Fault-Tolerant Topology and Routing Synthesis for IEEE Time-Sensitive Networking," in *Proc. of the 25th Int. Conf. on Real-Time Networks and Systems*, 2017.

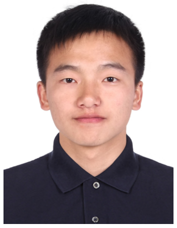
[19] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer, and K. Rothermel, "NeSTiNg: Simulating IEEE Time-sensitive Networking (TSN) in OMNeT++," in *Proc. of IEEE International Conference on Networked Systems*, 2019.

[20] D. Tamas-Selicean, P. Pop, and W. Steiner, "Design optimization of TTEthernet-based distributed real-time systems," Real-Time Systems, vol. 51, no. 1, pp. 1-35, 2015.



Luxi Zhao received the PhD in communication and information system from the Beihang University, Beijing, China, in 2017. She has been a postdoc at DTU Compute, Technical University of Denmark (DTU) from 2017 to 2019. She is currently a research fellow at department of Electrical and Computer Engineering, Technical University of Munich (TUM). Her main research interest concerns worst-case analysis and performance evaluation of deterministic real-time and safety-critical networks; runtime configuration and reconfiguration problems.

Prof. Paul Pop is a Professor of Cyber-Physical Systems at DTU Compute, Technical University of Denmark (DTU). He has received his Ph.D. degree in computer systems from Linköping University in 2003. His research is focused on developing methods and tools for the analysis and optimization of networked dependable cyber-physical systems. In this area, he has published over 130 peer-reviewed papers, three books, and seven book chapters. He has served as a technical program committee member on several conferences, such as DATE and ESWEEK. He has received the Best Paper Award at DATE 2005, RTIS 2007, CASES 2009, MECO 2013, and DSD 2016. He is the Chairman of the IEEE Danish Chapter on Embedded Systems. He is the coordinator of the Nordic University Hub on Industrial IoT and of the European Training Network on Fog Computing for Robotics and Industrial Automation.



Zijie Gong received the B.S. degree in electronic and information engineering from Beihang University, Beijing, China, in 2018. He is currently pursuing the M.S. degree at the School of Electronic Information Engineering, Beihang University, China. His main research interests include timing analysis and real-time networks.

Bingwu Fang received the B.S. degree in electronic and information engineering from Beijing Jiaotong University, Beijing, China, in 2014. He is currently pursuing the M.S. degree in School of Electronic and Information Engineering at Beihang University, Beijing, China. His research interests include real-time scheduling optimization, performance evaluation and real-time communication systems.