IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 11, 2018, accepted July 11, 2018, date of publication July 23, 2018, date of current version August 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2858767

Worst-Case Latency Analysis for IEEE 802.1Qbv Time Sensitive Networks Using Network Calculus

LUXI ZHAO[1], PAUL POP[1], AND SILVIU S. CRACIUNAS[1,2]
[1]Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Copenhagen, Denmark
[2]TTTech Computertechnik AG, 1040 Vienna, Austria

Corresponding author: Luxi Zhao (luxzha@dtu.dk)

ABSTRACT Distributed safety-critical applications in industrial automation, aerospace, and automotive, require worst-case end-to-end latency analysis for critical communication flows in order to prove their correct behavior in the temporal domain. With the advent of time sensitive networks (TSNs), distributed applications can be built on top of standard Ethernet technologies without sacrificing real-time characteristics. The time-based transmission selection and clock synchronization mechanism defined in the TSN enable the real-time transmission of frames based on a global schedule configured through so-called gate control lists (GCLs). This paper has an enhancement of allowing a mixture of the priority-based scheduling and time-triggered, which expand the solution space for the GCLs. Then, it is necessary to analyze the latency bounds for the critical traffic in the TSN network. In this paper, we start from the assumption that the GCLs, i.e., the communication schedules, and the traffic class (priority) assignment for critical flows are given for each output port and derive, using network calculus, an analysis of the worst-case delays that individual critical flows can experience along the hops from sender to receiver(s). Our method can be employed for the analysis of the TSNs where the GCLs have been created in advance, as well as for driving the GCL synthesis that explores a larger solution space than previous methods, which required a complete isolation of transmission events from different traffic classes. We validate our model and analysis by performing experiments on both synthetic and real-world use-cases, showing the scalability of our implementation as well as the impact of certain GCL properties (gate overlapping and traffic class assignments) on the worst-case latency of critical communication flows.

INDEX TERMS Performance analysis, delay, TSN, deterministic Ethernet, network calculus.

I. INTRODUCTION

Distributed safety-critical applications, like those found in the aerospace domain, require certification evidence for the correct real-time behavior of critical communication. More recently, other domains, like automotive and industrial automation, are also moving towards providing real-time guarantees for communication traffic in distributed applications built on top of standard Ethernet technologies. Driven by industrial automation, there have been increasing efforts to extend the Ethernet protocol with capabilities that enable some form of real-time communication. One of these efforts is currently ongoing within the IEEE 802.1 Time Sensitive Networking (TSN) task group [1] that defines amendments to the IEEE 802 standard enabling time-sensitive capabilities ranging from time-based filtering and policing, to clock synchronization and frame preemption as well as functionality for time-based dispatching of frames in switches and end-systems.

Within the TSN amendments, safety-critical communication is configured via so-called time-triggered (TT) flows (also called time-sensitive streams), where a TT flow is a communication requirement from a talker to one or more listeners. The IEEE 802.1Qbv [2] amendment introduces time-based gates at the output ports that bind the transmission of frames from the egress queues (traffic classes) to a configured periodic schedule called the *Gate Control List* (GCL). A timed gate can be in one of two states, either *open* or *closed*, at a given time. When the gate opens, frames from the respective queue are forwarded for transmission in first-in first-out (FIFO) order to a priority selection mechanism that determines which of the frames is forwarded to the connected physical link. Hence, if two gates are opened at the same time, their traffic class priority enforces that the higher priority frame is selected for transmission delaying the other lower priority traffic. This time-based transmission selection only works on a network level if an appropriate synchronization

mechanism, in this case IEEE 802.1ASrev [1], is in place that provides a global clock reference to all devices participating in the communication within a bounded precision.

Using mainly these two amendments, critical traffic can be scheduled [3]–[5] in a more deterministic fashion allowing temporal isolation and compositional system design for time-triggered flows that are scheduled end-to-end. GCLs can be created either to provide a periodic bandwidth guarantee and servicing on the level of traffic classes or to enable a fully-deterministic transmission pattern for individual flows. In the first case, the opening and closing of the timed gates determine the temporal behavior of entire traffic classes (communication priorities) while in the second case they determine the temporal behavior of individual flows regardless of the communication priority. For the second case, Craciunas *et al.* [3] presented a method that, given a split of the queues on the output ports into two categories (scheduled and non-scheduled), generates a correct schedule with zero jitter as well as deterministic end-to-end latencies for critical (time-triggered) communication flows. The deterministic communication behavior is achieved by enforcing a complete isolation of critical flows from each other which eliminates the need for a worst-case delay analysis at the expense of potentially reducing the solution space for TSN networks significantly.

In this work, we start from the assumption that the GCL, i.e. the communication schedule, and the traffic class (priority) assignment for TT flows are given for each output port and calculate the worst-case delays that individual critical flows can experience along the hops from talkers to listeners. As opposed to [3], we make no assumption on how the GCLs have been created or whether they provide a complete isolation of critical flows from each other, thus covering the most generic use-case for TSN networks. Hence, we are interested in providing a worst-case delay analysis for TT flows in order to verify the real-time latency requirements of critical communication. Our method can also be used in conjunction with a GCL optimization approach (e.g. [4]) to drive the optimization towards solutions where the worst-case latency of flows is determined by the presented method. Hence, the analysis can be used as a feedback mechanism to drive the GCL synthesis thus increasing the solution space, i.e., use-cases which are not schedulable with the isolation approach from [3] may become schedulable if windows are allowed to overlap since the strict isolation condition between critical flows from [3] is not necessary any more.

Traditionally, an analysis of worst-case end-to-end timing behavior in non-time-sensitive network nodes (i.e., via standard priority-based enforcement) has been achieved through methods like network calculus [6], [7] or the more recent trajectory approach [8]. In [9]–[11], a network calculus-based analysis is presented that computes the latency bounds of rate-constrained (RC) flows in TTEthernet where the deterministic behavior of TT flows is given. Other works, such as [12] and [13], propose a timing analysis using network calculus for TDMA-based networks under different scheduling policies, i.e. FIFO, fixed-priority (FP) and weighted round robin (WRR). Recently, Zhao *et al.* [14] have provided an analysis for TSN networks on the level of non-critical AVB flows considering the impact on delays introduced by the time-aware gates defined in IEEE 802.1Qbv. The timing analysis of critical TT traffic has been addressed in [15]. However, it just assumes a typical situation that GCLs for critical TT traffic are non-overlapping. Thus every critical traffic class has exclusive link access during its gate open, i.e. without interference by critical traffic with higher or lower priority. To the best of our knowledge, this is the first work that provides a latency analysis for critical TT flows of different priorities in the generic case in which there is no strict isolation within the critical traffic classes and no constraint on the design of GCLs, i.e., allowing the overlapping case. Therefore, the main contribution of the paper is proposing an network calculus-based method to be used for analysis and certification of critical communication in TSN networks as well as for more generic TSN GCLs that explore a larger solution space than previous methods.

We start by introducing the relevant portions of the 802.1Qbv mechanism together with the motivation for our work in Section II. We then present the system model in Section III followed by an overview of network calculus in Section IV. In Section V we derive both the aggregate arrival curve for competing TT flows as well as the service curve supplied by the device. In Section VI we evaluate our implementation, followed by concluding remarks in Section VII.

II. 802.1Qbv FUNCTIONALITY

Here, we briefly introduce the specifics of the IEEE 802.1Qbv sub-standard and explain the motivation for our approach. We remind the reader that critical traffic in TSN networks is configured via so-called time-triggered (TT) flows (also called time-sensitive streams), where a TT flow consists of one or multiple routed frames of a given size that are sent from a talker to one or more listeners through intermediary switches.

Fig. 1 depicts a simplified representation of the internals of a 802.1Qbv-capable switch with three input ports and one output port. Incoming traffic is first filtered through a switching fabric and redirected to the desired output port

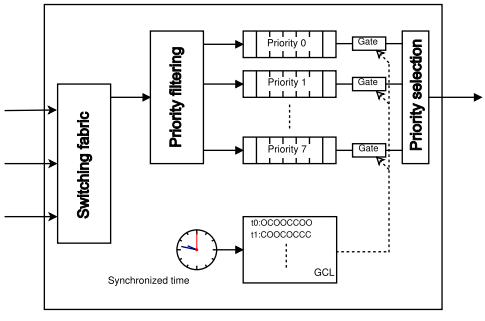


FIGURE 1. Simplified view of a 802.1Qbv-capable switch.

based on the routing table configured in the switch. A priority filter then determines to which of the 8 priority queues of the chosen output port a certain flow is redirected based on either the priority code point (PCP) of the IEEE 802.1Q header or the stream identification function defined in IEEE 802.1Qci. At each output port, there are 8 queues buffering frames in FIFO order until their transmission is enabled through the timed gate associated with the queue. The GCL enforces at which time instants traffic is forwarded to the output physical port and at which times it is buffered and not allowed to go through. If multiple queues are opened at the same time, the forwarding of frames to the physical link is determined by the queue priority in the priority selection block.

Gate Control Lists, i.e. schedules, can serve two purposes in TSN networks. They can either control the behavior of critical frames on the level of traffic class or on the level of flows. In the first case, bandwidth reservation or time-based shaping can be enforced through the schedule without controlling the exact temporal behavior of individual frames but rather of an entire traffic class (priority). Especially in legacy systems, TSN offers the possibility to shape traffic using a synchronized network time as opposed to classical mechanisms, such as the Credit-Based Shaper. In this case frames might be delayed not only by higher and lower priority frames whose queue gate is opened at the same time but also by frames of different flows that have the same priority.

In the second case, a scheduling algorithm [3] creates schedules for time-triggered flows such that zero jitter as well as deterministic end-to-end latencies are guaranteed for critical traffic. In a first step, the queues are assigned to be either scheduled (TT-queues) or unscheduled (priority-queues). Critical traffic is isolated from non-critical traffic into the scheduled queues which are always the highest priority of all existing queues. Unscheduled queues follow a strict priority policy, meaning that the arbitration is purely based on queue priority and the gates of the queues are opened at the same time for all priority queues. For critical traffic there are several scenarios that have to be considered in order to guarantee timely behavior. On one hand, if two scheduled queues are opened at the same time, individual frames of the higher priority flows may delay the frames of the lower-priority flows leading to increased latency and jitter. On the other hand, if for example two flows are scheduled to arrive at a switch from different sources at roughly the same time, individual frames of these flows may be placed in the same queue at runtime in a non-deterministic way due to frame loss or synchronization errors. If for example two frames arrive at roughly the same time at one switch, due to synchronization errors that change at runtime (within a known upper bound), the queueing order of the frames may be different in different period instances. Hence, when the gate opens for the respective queue, the sending order of the two frames depends on the current queueing order thus introducing non-determinism. It is important to note that the GCL controls the timing of the opening and closing of the queue, not the order of frames in the queue. If the order of frames in the queue, i.e. the queue state, is not deterministic, the timely behaviour of the two flows may oscillate in each periodic instance allowing frames from the two flows to delay each-other and thus accumulate jitter for the overall end-to-end transmission.

The GCL synthesis algorithm in [3] enforces that no two scheduled queues of the same output port are opened at the same time, essentially meaning that the priority is overruled by the schedule in order to avoid the first source of non-determinism mentioned above. Moreover, in order to achieve deterministic behavior and avoid the second scenario, the state of each queue has to be deterministic in order to isolate critical flows from each-other. While the work [3] offers a way to enforce a deterministic behavior for critical traffic, it has to impose restrictive conditions of isolation between critical flows in order not to necessitate a delay analysis of competing TT flows. This may lead to a significantly reduced solution space for TSN networks. Hence, in order to maximize the solution space, GCL synthesis must allow some degree of non-determinism in the queues, i.e., relax the isolation constraint, while still provide guarantees for critical traffic. In order to achieve this, an analysis of the worst-case delay in the scheduled queues is necessary.

In this work we want to address the most generic case and propose an analysis of worst-case delays for time-triggered flows when the opening and closing of scheduled queues are allowed to overlap and frames of different critical flows are allowed to interleave in the same queue. Hence, our starting point is that the GCLs are given and there is no mutually exclusive restriction on the events for scheduled queues and, furthermore, there is no restriction on the isolation of flows in different queues. This analysis can be used to give worst-case estimations in legacy systems that have been enhanced by GCL-based bandwidth reservations, as a verification step in a generic GCL synthesis algorithm that maximizes the solution space, or it can be used as a analysis step for optimizing existing GCLs, acting as feedback to the GCL optimization algorithm.

III. SYSTEM MODEL

A. ARCHITECTURE MODEL

A TSN network is a graph consisting of a set of nodes which can be either end systems (ES) or switches (SW), connected through physical links. The links are assumed to be full duplex, allowing communication in both directions. An example is presented in Fig. 2, where we have 4 end-systems, ES_1 to ES_4, and 3 switches, SW_1 to SW_3.

The topology of a TSN network 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 ($\boldsymbol{ES}$) and switches ($\boldsymbol{SW}$), and $\boldsymbol{E}$ is the set of physical links. For Fig. 2, $\boldsymbol{V} = \boldsymbol{ES} \bigcup \boldsymbol{SW} = \{ES_1, ES_2, ES_3, ES_4\} \bigcup \{SW_1, SW_2, SW_3\}$, and the physical links are depicted with black, double arrows. A dataflow link $dl_i = [v_a, v_b] \in \boldsymbol{L}$, where $\boldsymbol{L}$ is the set of dataflow links in the

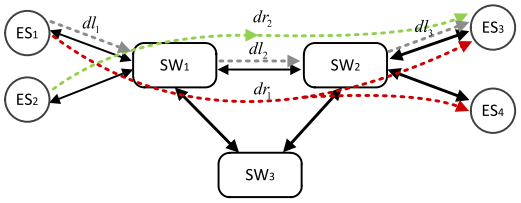


FIGURE 2. TSN network topology example.

network, is a directed edge from v_a to v_b, where v_a and $v_b \in \boldsymbol{V}$ can be ESes or SWs. The physical link rate is denoted as $dl_k \cdot C$. In this paper, we assume that all physical links have the same rate C. As there is only one output port for each dataflow link, dl_k can also refer to the output port h in v_a associated with the link to v_b. A dataflow routing $dr_k \in \boldsymbol{R}$ is an ordered sequence of dataflow links connecting a single source ES to one or more destination ESes. For example in Fig. 2, dr_1 connects the source end system ES_1 to the destination end systems ES_3 and ES_4, while dr_2 connects ES_2 to ES_3.

B. APPLICATION MODEL

The tasks of applications running in ESes communicate via flows, which have a single source and may have multiple destinations. While there may be non-critical communication in the network, in this paper we concentrate on the critical flows that have real-time requirements. We define $\tau = \bigcup_k \tau_{TT_k}$ as the set of all critical flows in the TSN network.

As mentioned, the TSN standard supports different priorities for critical TT flows. It is assumed that the priority P_m ($m \in [1, 8]$) for each critical TT flow has been decided by the system designer. Moreover, for each TT flow $\tau_{TT_k} \in \tau$, we know its frame size l_{TT_k}, the period p_{TT_k} in the source ES and the statically defined routing dr_{TT_k}.

IV. NETWORK CALCULUS BACKGROUND

We briefly introduce here, the necessary background of network calculus theory. Network calculus [16]–[18] is a theory designed for the deterministic performance analysis of network communication, such as the worst-case bounds on latencies and backlogs of flows transmitted within a network. In order to perform the analysis, appropriate arrival and service curves describing the behavior of flows and the availability of network nodes have to be constructed. Service and arrival curves are computed by means of the min-plus convolution operation of $(min, +)$ algebra [17].

An arrival curve $\alpha(t)$ models the arrival process $R(t)$ of a flow, where $R(t)$ represents the cumulative input function counting the total bits of the flow entering the network until time t. We say that $R(t)$ is constrained by $\alpha(t)$ if

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

where inf means infimum (greatest lower bound) and $\otimes$ is the notation of the min-plus convolution operation. A typical example of an arrival curve is the *leaky bucket* model described by

$$\alpha_{\sigma,\rho}(t) = \begin{cases} \rho t + \sigma, & t \geq 0 \\ 0, & t < 0, \end{cases}$$

where σ represents the maximum burst of the flow and ρ is the upper bound of the long-term average rate of the flow.

A service curve $\beta(t)$ models the processing capability of the available resource. Assume that $R^*(t)$ is the departure process, which is the cumulative output function that counts the total bits of the flow that exit the network node up to time t. We say that the network node offers the service curve $\beta(t)$ for the flow if

$$R^*(t) \geq \inf_{0 \leq s \leq t} \{R(s) + \beta(t-s)\} = (R \otimes \beta)(t). \qquad (2)$$

A typical example of the service curve is the rate-latency service curve described by

$$\beta_{R,T}(t) = R[t-T]^+,$$

where R represents the service rate, T represents the service latency and the notation $[x]^+$ is equal to x if $x \geq 0$ and 0 otherwise.

Let us assume that a flow constrained by the arrival curve $\alpha(t)$ traverses the network node offering the service curve $\beta(t)$. Then, the latency experienced by the flow in the network node is bounded by the maximum horizontal deviation between the graphs of two curves $\alpha(t)$ and $\beta(t)$

$$h(\alpha, \beta) = \sup_{s \geq 0} \{\inf \{\tau \geq 0 \mid \alpha(s) \leq \beta(s+\tau)\}\}, \qquad (3)$$

where sup means supremum (least upper bound). Consider a flow constrained by the leaky bucket $\alpha_{\sigma,\rho}(t)$ and served in a node with the rate-latency service curve $\beta_{R,T}(t)$. The worst-case latency is shown using the grey double arrow labelled with $h(\alpha, \beta)$ in Fig. 3. The worst-case end-to-end delay of the flow is the sum of latency bounds in network nodes along its routed virtual link path.

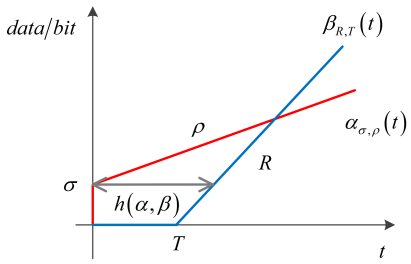


FIGURE 3. A simple analysis example by using network calculus.

V. WORST-CASE ANALYSIS FOR CRITICAL TRAFFIC

In order to calculate the worst-case latency of a critical flow in an output port of a node, it is necessary to obtain the aggregate arrival curve for competing critical flows on that port, and the service curve supplied for them by the device. Then the worst-case end-to-end delay (WCD) of the critical flow can

be obtained by disseminating the latency bounds along its routed path. The notations appeared in this section are given in V-E.

In general, it is assumed that there are n ($1 \leq n \leq 8$) queues for critical traffic respectively with different priority levels P_m ($1 \leq m \leq n$). Critical flows of different priority levels P_m are placed in the corresponding queues Q_{P_m}. Frames in each queue follow the first-in-first-out (FIFO) order, and frames have higher priority in the queue Q_{P_m} than in the queue $Q_{P_{m+1}}$. Moreover, there is a gate G_{P_m} for the queue Q_{P_m} controlled by the GCL with the open ($G_{P_m}(t) = 1$) and closed ($G_{P_m}(t) = 0$) states, and frames are allowed to be forwarded only when the associated gate is open. It is assumed that the open-close cycle of the gate G_{P_m} is T_{P_m} and the length of open window in an open-close cycle is L_{P_m}, as shown in Fig. 4.

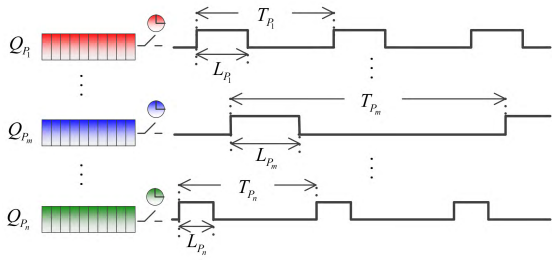


FIGURE 4. Open-close cycle and open windows of each queue gate.

The 802.1Qbv [2] standard defines a lookahead mechanism for each traffic class to check whether there is enough time to send the entire frame before the closing time of the gate. If not, the frame cannot be forwarded until the next open window, and there will be an idle time (guard band) at the end of the current open window. In the worst-case, the guard band is no longer than the Ethernet Maximum Transmission Unit (MTU) of 1500 bytes. In addition, if gates are open at the same time, we assume the non-preemption policy (i.e. no IEEE 802.1Qbu support), which implies that the critical frame already on transmission cannot be interrupted by a higher priority frame. Note that the service resource during an open window of Q_{P_m} may not be dedicated to P_m traffic due to the possibility of overlapping with open windows of other priority queues.

In the following we are interested to derive the service curve for each priority of critical traffic, by deriving the guaranteed service in each open window and the worst-case waiting time before each guaranteed time slot.

A. SERVICE CURVE FOR CRITICAL TRAFFIC WITH PRIORITY P_m

The service in one node for the critical traffic of priority P_m is different from the TDMA service. The service of TDMA for each traffic class is mutually exclusive and dedicated. Therefore, the lower bound (guaranteed service) offered TDMA time slot for the traffic is repeated with the fixed cycle length. However, as mentioned above, the service time slot controlled by GCLs in TSN will not be dedicated for P_m traffic if overlapping occurs.

Then, in order to obtain the service curve for the P_m traffic, it is necessary to obtain the time slot length $\overline{L}_{P_m}$ of guaranteed service in each open window of Q_{P_m} of P_m. It is not only related to the worst-case overlapping with the higher and lower priority critical traffic, but also to the maximum guard band before each open window of Q_{P_m}, for example, in Fig. 5. We also need to analyze the maximum waiting time S_{P_m} when the first frame of a backlog period of P_m obtains the service from its corresponding time slot. It is related to maximum frame size in Q_{P_m} and the worst-case overlapping with other priority traffic, as shown in Fig. 5 for example. Both the guaranteed time slot $\overline{L}_{P_m}$ in an open window and the maximum waiting time S_{P_m} will be separately discussed in Sect. V-B and Sect. V-C.

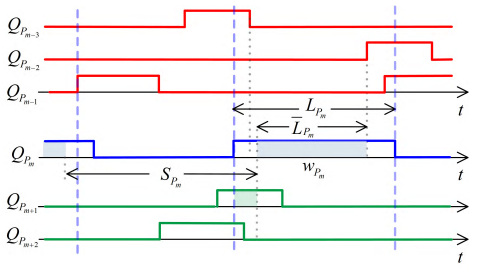


FIGURE 5. Guaranteed time slot in an open window.

Moreover, the time slot length $\overline{L}_{P_m}$ of guaranteed service is variable in different open windows of Q_{P_m}, due to different overlapping situations with higher and lower priority queues, as shown in Fig. 6 marked with $\overline{L}^1_{P_m}$, $\overline{L}^2_{P_m}$ and $\overline{L}^3_{P_m}$. Note that for a given GCL of an output port, the overlapping relations are repeated with the hyperperiod T_{GCL}, which is the Least Common Multiple (LCM) of open-close cycles T_{P_m} ($1 \leq m \leq n$) for all the priority queues of the critical traffic. Therefore, there is a limited number of cases of length $\overline{L}_{P_m}$ of guaranteed time slot for P_m traffic, and it is assumed to be denoted with N_{P_m}. For example N_{P_m} is 3 when $T_{P_{m-1}} = 6ms$, $T_{P_m} = 2ms$, $T_{P_{m+1}} = 3ms$ and thus the hyperperiod $T_{GCL} = 6$ in Fig. 6. Note that the maximum waiting time S_{P_m} is also variable depending on the reference open window.

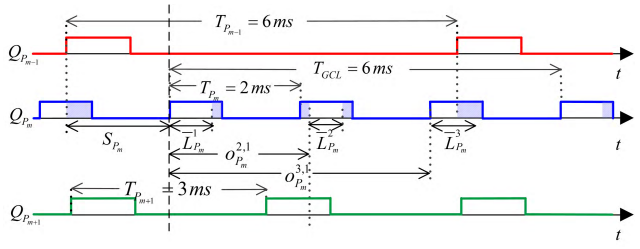


FIGURE 6. Guaranteed time slots for P_m.

As we can see, the guaranteed service for P_m traffic loses the periodicity (T_{P_m}). In order to represent the positional

relationship of adjacent guaranteed time slot, we define the relative offset $o_{P_m}^{j,i}$ ($j \in [i+1, i+N_{P_m}-1]$), which is the time interval between the starting time of the ith and jth guaranteed time slots for P_m traffic, by taking the ith guaranteed time slot as the reference. For example in Fig. 6, $o_{P_m}^{2,1}$ and $o_{P_m}^{3,1}$ respectively represent the relative offsets between $\overline{L}_{P_m}^1$ and $\overline{L}_{P_m}^2$ and between $\overline{L}_{P_m}^1$ and $\overline{L}_{P_m}^3$, by taking the guaranteed service window with the length $\overline{L}_{P_m}^1$ as the reference. Note that $o_{P_m}^{j,i}$ is equal to 0 if $j = i$.

Theorem 1: A possible service curve $\beta_{P_m}^i(t)$ for the critical traffic of priority P_m, considering the guaranteed time slot $\overline{L}_{P_m}^i$ ($i = 1, \ldots, N_{P_m}$) as the benchmark, is given by

$$\beta_{P_m}^i(t) = \sum_{j=i}^{i+N_{P_m}-1} \beta_{P_m}^{j,i}(t), \tag{4}$$

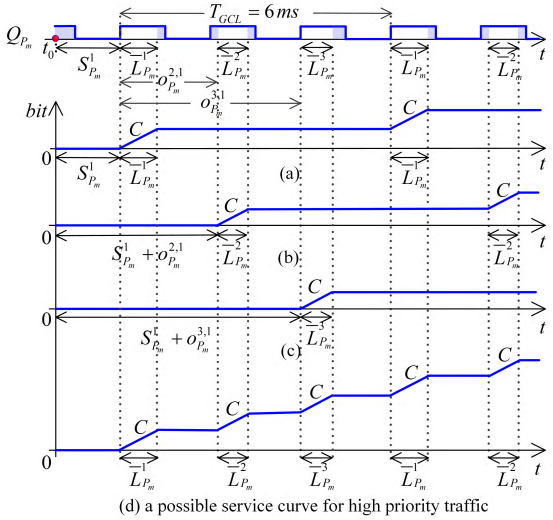
where

$$\beta_{P_m}^{j,i}(t) = \beta_{T_{GCL},\overline{L}_{P_m}^j}(t + T_{GCL} - \overline{L}_{P_m}^j - S_{P_m}^i - o_{P_m}^{j,i}), \tag{5}$$

and $\beta_{T,L}(t)$ is the classic service curve of fluid flow models combined with the TDMA protocol [12],

$$\beta_{T,L}(t) = C \cdot max(\lfloor \frac{t}{T} \rfloor L, t - \lceil \frac{t}{T} \rceil (T - L)). \tag{6}$$

Proof: By taking the guaranteed time slot $\overline{L}_{P_m}^i$ as the benchmark, it means that the first frame of the backlog period of P_m will obtain the service starting from the time slot $\overline{L}_{P_m}^i$. Then we separately consider N_{P_m} sequences of periodic guaranteed time slots, which are separately repeated according to the hyperperiod T_{GCL}, to derive the service curve, for example in Fig. 7 (a) to (c).



(d) a possible service curve for high priority traffic

FIGURE 7. Worst-case service scenario for high priority traffic based on a benchmark.

Then, for a periodic sequence of service time slots of which the length is $\overline{L}_{P_m}^j$ ($i \le j \le i + N_{P_m} - 1$), the service for P_m traffic cannot be guaranteed in any time interval $0 \le \Delta = t - t_0 < S_{P_m}^i + o_{P_m}^{j,i}$. However, it can guarantee the service of $C \cdot (\Delta - S_{P_m}^i - o_{P_m}^{j,i})$ in any time interval $S_{P_m}^i + o_{P_m}^{j,i} \le \Delta \le S_{P_m}^i + o_{P_m}^{j,i} + \overline{L}_{P_m}^j$.

In addition, as the guaranteed time slot $\overline{L}_{P_m}^j$ is repeated with T_{GCL}, the service during any time interval $S_{P_m}^i + o_{P_m}^{j,i} + \theta \cdot T_{GCL} + \overline{L}_{P_m}^j \le \Delta < S_{P_m}^i + o_{P_m}^{j,i} + (\theta+1) \cdot T_{GCL}$ ($\forall \theta \in \boldsymbol{N}$) cannot be guaranteed for P_m traffic, while the service of $C \cdot (\Delta - (S_{P_m}^i + o_{P_m}^{j,i} + (\theta+1) \cdot T_{GCL}))$ in any time interval $S_{P_m}^i + o_{P_m}^{j,i} + (\theta+1) \cdot T_{GCL} \le \Delta < S_{P_m}^i + o_{P_m}^{j,i} + (\theta+1) \cdot T_{GCL} + \overline{L}_{P_m}^j$ can be guaranteed for P_m traffic. Thus, the service curve for a sequence of periodic time slots with the length $\overline{L}_{P_m}^j$ can be given by

$$\beta_{P_m}^{j,i}(t) = \beta_{T_{GCL},\overline{L}_{P_m}^j}(t + T_{GCL} - \overline{L}_{P_m}^j - S_{P_m}^i - o_{P_m}^{j,i}),$$

where $S_{P_m}^i + o_{P_m}^{j,i}$ is the relative offset from the beginning time t_0 of the backlog period to the starting time of the guaranteed time slot $\overline{L}_{P_m}^j$, and $\beta_{T_{GCL},\overline{L}_{P_m}^j}(t)$ is the classic TDMA service curve.

Then, a possible service curve for P_m traffic by taking the guaranteed time slot $\overline{L}_{P_m}^i$ as the benchmark, is derived from the sum of service curves of each periodic sequence consisting time slots with length $\overline{L}_{P_m}^j$ ($i \le j \le i + N_{P_m} - 1$),

$$\beta_{P_m}^i(t) = \sum_{j=i}^{i+N_{P_m}-1} \beta_{P_m}^{j,i}(t),$$

as shown for example in Fig. 7 (d). ■

Up until now we have derived N_{P_m} possible service curves $\beta_{P_m}^i(t)$ ($i = 1, \ldots, N_{P_m}$) for P_m traffic by considering different guaranteed time slots in the hyperperiod as benchmarks. Then the service curve $\beta_{P_m}(t)$ for P_m traffic is the worst case of all the possible service curves, i.e., the lower envelope of $\beta_{P_m}^i(t)$,

$$\beta_{P_m}(t) = \min_{1 \le i \le N_{P_m}} \left\{ \beta_{P_m}^i(t) \right\}. \tag{7}$$

B. GUARANTEED TIME SLOT IN AN OPEN WINDOW

In this subsection, we are interested to analyze the guaranteed time slot $\overline{L}_{P_m}^i$. It is assumed that $t_{P_m}^{o,i}$ and $t_{P_m}^{c,i}$ are respectively the beginning and end times of the open window $w_{P_m}^i$, for example in Fig. 8. In addition, the overlapping situations with other priority queues are known and given by the GCL configuration.

1) EFFECT FROM LOWER PRIORITY TRAFFIC

As we assume the non-preemption policy among different traffic class, once overlapping of open windows occurs, a frame from lower priority P_{m^+} ($m+1 \le m^+ \le n$) that has been already on transmission cannot be interrupted by the frame queued in Q_{P_m} when the gate of Q_{P_m} is open, even though P_m frame has higher priority. For example in Fig. 8, during the backlog period of P_m traffic, the transmission

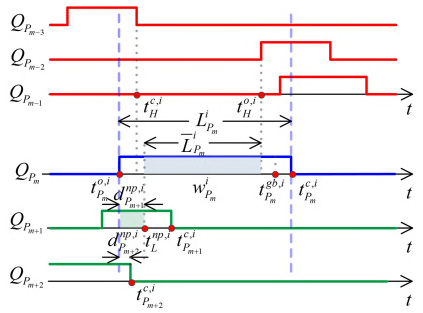


FIGURE 8. Guaranteed service time slot $\overline{L}^i_{P_m}$ in an open window $w^i_{P_m}$.

of P_m frames are blocked before $t^{o,i}_{P_m}$ due to closed gate ($G_{P_m} = 0$) of Q_{P_m}. At the time $t^{o,i}_{P_m}$, if a lower priority frame from Q_{m+1} or Q_{m+2} is already on transmission, which is absolutely possible according to the fact that its corresponding gate is open before $t^{o,i}_{P_m}$, then frames waiting in the Q_{P_m} still cannot be forwarded due to the non-preemption policy.

It means that, if the gate for a lower priority queue opens before $t^{o,i}_{P_m}$, and closes after, i.e., $G_{P_{m^+}}(t^{o,i}_{P_m}) = 1$, there could be a non-preemption latency for P_m traffic at the beginning of the corresponding open window $w^i_{P_m}$. Now we assume that $t^{c,i}_{P_{m^+}}$ $(m+1 \le m^+ \le n)$ is the nearest time when the gate $G_{P_{m^+}}$ is closed no earlier than $t^{o,i}_{P_m}$, i.e.,

$$t^{c,i}_{P_{m^+}} = \inf_{t \ge t^{o,i}_{P_m}} \left\{ G_{P_{m^+}}(t) = 0 \right\}.$$

On the one hand, due to the lookahead mechanism for the same traffic class, the non-preemption frame from lower priority P_{m^+} must finish the transmission before its queue gate closes. On the other hand, there is at most one non-preemption frame from lower priority queues causing the latency for P_m traffic in the open window $w^i_{P_m}$. Therefore, the worst-case non-preemption latency for the P_m traffic in the open window $w^i_{P_m}$ caused by a lower priority frame from queues $Q_{P_{m^+}}$ can be given by

$$d^{np,i}_{P_{m^+}} = \begin{cases} min\left\{ \dfrac{l^{max}_{P_{m^+}}}{C}, t^{c,i}_{P_{m^+}} - t^{o,i}_{P_m} \right\}, & G_{P_{m^+}}(t^{o,i}_{P_m}) = 1 \\ 0 & G_{P_{m^+}}(t^{o,i}_{P_m}) = 0, \end{cases} \tag{8}$$

where $l^{max}_{P_{m^+}}$ is the maximum frame size of lower priority flows of which open windows are overlapping with $w^i_{P_m}$, as shown in Fig. 8 for example.

Then by considering all the lower priority critical traffic, we obtain the worst-case non-preemption latency for the beginning of the open window $w^i_{P_m}$ as follows

$$d^{np,i}_L = \max_{m+1 \le m^+ \le n} \left\{ d^{np,i}_{P_{m^+}} \right\}. \tag{9}$$

Then we define the starting time of guaranteed time slot $\overline{L}^i_{P_m}$ for P_m traffic in $w^i_{P_m}$ by just considering the effect of lower priority traffic,

$$t^{np,i}_L = d^{np,i}_L + t^{o,i}_{P_m}. \tag{10}$$

Note that, during the backlog period of P_m, non-preemption latency from lower priority P_{m^+} $(m+1 \le m^+ \le n)$ could appear at the beginning of each open window $w^i_{P_m}$ $(i = 1, \ldots, N_{P_m})$ for the queue Q_{P_m}, if at this moment open windows of P_m and P_{m^+} are overlapping. This case differs from the purely priority driven scheduling, of which the resource is always occupied by higher priority traffic during the backlog period. However, in this paper, each priority queue is also controlled by the alternating open-close states of gates, so that the higher traffic can be blocked more than once by a lower priority frame even during the backlog period.

2) EFFECT OF THE LOOKAHEAD MECHANISM

At the end of the open window $w^i_{P_m}$, due to the lookahead mechanism, the maximum idle time (guard band) appears when a maximum-sized frame of P_m arrives at the instant when the leftover time before the gate G_{P_m} closed is slightly less than its required transmission time. Thus, the maximum guard band latency is equal to

$$d^{gb}_{P_m} = \frac{l^{max}_{P_m}}{C}, \tag{11}$$

where $l^{max}_{P_m}$ is the maximum frame size of all the flows with priority P_m.

We then assume that the guaranteed time slot for the P_m traffic in the open window $w^i_{P_m}$ ends with

$$t^{gb,i}_{P_m} = t^{c,i}_{P_m} - d^{gb}_{P_m}, \tag{12}$$

if consider only the effect of the lookahead mechanism, as shown in Fig. 8.

3) EFFECT FROM HIGHER PRIORITY TRAFFIC

The guaranteed service for P_m traffic during the open window $w^i_{P_m}$ is also related to the maximum service for higher priority traffic P_{m^-} $(1 \le m^- \le m-1)$. In the worst-case, the arrival characteristics of P_{m^-} traffic are shaped by the periodic open windows $w_{P_{m^-}}$ defined by GCLs.

Our guaranteed service analysis for P_m does not make any assumptions on the individual frames of higher priority traffic but considers only the interference on the level of the open windows $w_{P_{m^-}}$. Considering individual higher-priority frames would result in less pessimistic results for the P_m traffic, but requires complete knowledge of the traffic present in the system. Since we want to maintain the flexibility to support legacy systems, in which only the GCL is known but some of the existing traffic is unknown, we choose to study the interference on the level of open windows defined by the GCLs rather than frames. An extension that considers individual frames is subject for future work.

Hence, when considering the interference effects from the level of higher priority open windows, the service cannot be guaranteed for P_m traffic during the overlap intervals of P_{m^-} and P_m. These overlap intervals of open windows of P_{m^-} and P_m can be of the types shown in Fig. 9. The case presented in Fig. 9 (c) can be seen as a combination of the overlapping cases shown in Fig. 9 (a) and (b), where the two

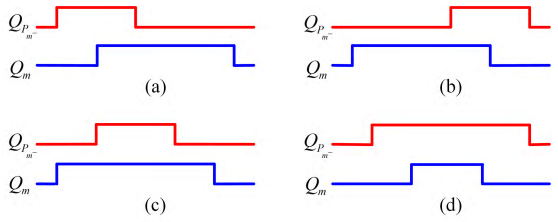


FIGURE 9. Overlapped cases of open windows between P_{m^-} and P_m.

open windows for Q_m have 0 distance between each other. The case in Fig. 9 (d) shows that the whole open window $w_{P_{m^-}}$ from higher priority prevents w_{P_m} from transmitting, meaning that the guaranteed time slot for the P_m traffic in the open window w_{P_m} equals to 0. We hence concentrate our analysis on the overlapping cases shown in Fig. 9 (a) and (b).

It is assumed that $t^{c,i}_{P_{m^-}}$ is the earliest time no earlier than $t^{o,i}_{P_m}$ with the closed state of the gate $G_{P_{m^-}}$, i.e.,

$$t^{c,i}_{P_{m^-}} = \inf_{t \geq t^{o,i}_{P_m}} \left\{ G_{P_{m^-}}(t) = 0 \right\}, \tag{13}$$

and $t^{o,i}_{P_{m^-}}$ is the latest time no later than $t^{c,i}_{P_m}$ with the closed state of the gate $G_{P_{m^-}}$, i.e.,

$$t^{o,i}_{P_{m^-}} = \sup_{t \leq t^{c,i}_{P_m}} \left\{ G_{P_{m^-}}(t) = 0 \right\}. \tag{14}$$

Thus, during the interval $[t^{c,i}_{P_{m^-}}, t^{o,i}_{P_{m^-}}]$, the P_m traffic has no interference from higher priority traffic of P_{m^-}.

Then, by just considering the effect of all the higher priority traffic, the guaranteed service for the P_m traffic in the open window $w^i_{P_m}$ starts from

$$t^{B,i}_H = \max_{1 \leq m^- \leq m-1} \left\{ t^{c,i}_{P_{m^-}} \right\}, \tag{15}$$

and ends with

$$t^{E,i}_H = \min_{1 \leq m^- \leq m-1} \left\{ t^{o,i}_{P_{m^-}} \right\}. \tag{16}$$

4) COMPREHENSIVE EFFECT CONSIDERATIONS

Hence, as mentioned above, the guaranteed time slot $\overline{L}^i_{P_m}$ for the P_m traffic in the open window $w^i_{P_m}$ is related to the non-preemption effect $t^{np,i}_L$ at the beginning of $w^i_{P_m}$, the guard band effect $t^{gb,i}_{P_m}$ at the end of $w^i_{P_m}$, and the interval $[t^{B,i}_H, t^{E,i}_H]$ with no interference from higher priority traffic.

If $t^{np,i}_L > t^{B,i}_H$, it means that in the worst-case the latency for P_m traffic caused by a non-preemption frame from lower priority is greater than caused by the service for the higher priority traffic, and vice versa. Therefore, the starting time of the guaranteed time slot $\overline{L}^i_{P_m}$ for the P_m traffic in the open window $w^i_{P_m}$ is

$$t^{B,i}_{P_m} = max \left\{ t^{np,i}_L, t^{B,i}_H \right\}. \tag{17}$$

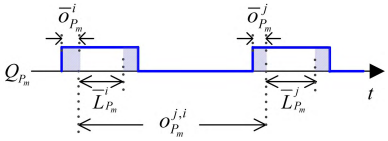


FIGURE 10. Relations of relative offsets $\overline{o}^i_{P_m}$ and $o^{j,i}_{P_m}$.

If $t^{gb,i}_{P_m} < t^{E,i}_H$, it means that in the worst-case the latency for P_m traffic caused by the lookahead mechanism is greater than caused by the service for the higher priority traffic, and vice versa. Thus, the ending time of the guaranteed time slot $\overline{L}^i_{P_m}$ for the P_m traffic in the open window $w^i_{P_m}$ is

$$t^{E,i}_{P_m} = min \left\{ t^{gb,i}_{P_m}, t^{E,i}_H \right\}. \tag{18}$$

Note that due to the non-preemption of P_m traffic, its frame cannot be interrupted if it starts to be transmitted during $[t^{B,i}_{P_m}, t^{E,i}_{P_m}]$. Therefore, the lower bound of service offered for the P_m traffic is always larger than $l^{min}_{P_m}/C$ if $t^{E,i}_{P_m} > t^{B,i}_{P_m}$, where $l^{min}_{P_m}$ is the minimum frame size of all the flows with priority P_m. In summary, the length $\overline{L}^i_{P_m}$ of the guaranteed service slot in the open window $w^i_{P_m}$ is given by

$$\overline{L}^i_{P_m} = \begin{cases} max \left\{ t^{E,i}_{P_m} - t^{B,i}_{P_m}, \dfrac{l^{min}_{P_m}}{C} \right\}, & t^{B,i}_{P_m} < t^{E,i}_{P_m} \\ 0 & t^{B,i}_{P_m} \geq t^{E,i}_{P_m}. \end{cases} \tag{19}$$

Moreover, the relative offset of the guaranteed time slot $\overline{L}^i_{P_m}$ from the beginning of the open window $w^i_{P_m}$ can be given by

$$\overline{o}^i_{P_m} = t^{B,i}_{P_m} - t^{o,i}_{P_m}\Big|_{\overline{L}^i_{P_m} \neq 0}, \tag{20}$$

as shown in Fig. 10. Then, the relative offset $o^{j,i}_{P_m}$ between the starting time of the ith and jth guaranteed time slots, considering the ith open window $w^i_{P_m}$ as the reference, is

$$o^{j,i}_{P_m} = (j-i) \cdot T_{P_m} - \overline{o}^i_{P_m} + \overline{o}^j_{P_m}. \tag{21}$$

C. THE MAXIMUM WAITING TIME AT THE BEGINNING OF THE BUSY PERIOD

This subsection focuses on the analysis of the maximum waiting time S_{P_m} at the beginning of the backlog period of P_m. It happens when the first frame $f^0_{P_m}$ of the backlog period arrives at the instance when the remaining time in the open window before the gate closed is maximized but cannot be guaranteed for the service of the P_m traffic. Thus, $f^0_{P_m}$ will be delayed to the next adjacent guaranteed time slot. Different from the non-preemption in the previous section, the non-preemption latency may only appear at the beginning of an open window. However, at the beginning of the backlog period, the non-preemption effect can appear at any instant as long as the open windows of P_m and P^+_m $(m+1 \leq m^+ \leq n)$ are overlapping. Therefore, the maximum

waiting time S_{P_m} is related not only to the non-guaranteed service, as discussed in the previous section, but also to the additional non-preemption latency at the beginning of the backlog period.

It is assumed that the guaranteed time slot $\overline{L}^{i}_{P_m}$ for the P_m traffic is considered to the benchmark. It means that the earliest processing time for the first frame $f^0_{P_m}$ of P_m in the backlog period will obtain the service starting from $\overline{L}^{i}_{P_m}$. Then, we can find the previous adjacent guaranteed time slot $\overline{L}^{i-1}_{P_m}$ for P_m traffic, as shown in Fig. 11. If $f^0_{P_m}$ arrives at some instant during the interval $[t^{E,i-1}_{P_m}, t^{B,i}_{P_m}]$, the service cannot be guaranteed for it.

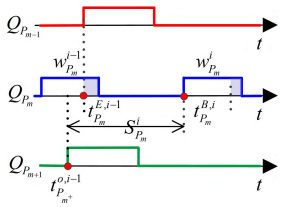


FIGURE 11. The maximum waiting time at the beginning busy period.

In addition, it is assumed that $t^{o,i-1}_{P_{m^+}}$ $(m+1 \le m^+ \le n)$ is the latest time no later than $t^{E,i-1}_{P_m}$ with the closed state of the gate $G_{P_{m^+}}$, i.e.,

$$t^{o,i-1}_{P_{m^+}} = \sup_{t \le t^{E,i-1}_{P_m}} \left\{ G_{P_{m^+}}(t) = 0 \right\}.$$

Hence, by considering all the lower priority critical traffic, the worst-case non-preemption latency at the beginning of the backlog period is,

$$d^{np,0}_{L} = \max_{m+1 \le m^+ \le n} \left\{ min \left\{ \frac{l^{max}_{P_{m^+}}}{C}, t^{E,i-1}_{P_m} - t^{o}_{P_{m^+}} \right\} \right\}. \quad (22)$$

In summary, the maximum waiting time $S^i_{P_m}$ at the beginning of the backlog period, by taking the guaranteed time slot $\overline{L}^{i}_{P_m}$ as the benchmark, can be given by

$$S^i_{P_m} = d^{np,0}_{L} + t^{B,i}_{P_m} - t^{E,i-1}_{P_m}, \quad (23)$$

where $t^{B,i}_{P_m}$ is the starting time of the guaranteed time slot $\overline{L}^{i}_{P_m}$, and $t^{E,i-1}_{P_m}$ is the ending time of the guaranteed time slot $\overline{L}^{i-1}_{P_m}$.

D. WORST-CASE END-TO-END LATENCY FOR CRITICAL TRAFFIC

From the previous section, we obtain the service curve $\beta^h_{P_m}(t)$ from (7) for each priority of critical traffic in an output port h of a node. If the aggregate arrival curve $\alpha^h_{P_m}(t)$ for the P_m traffic before the output port h is known, we can obtain the worst-case latency $D^h_{\tau_{TT_k}}$ of a critical TT flow $\tau_{TT_k} \in \tau^h_{P_m}$ waiting in the output port h along its routing, where $\tau^h_{P_m}$ is the set of flows of priority P_m in the output port h.

For any priority level of a critical flow τ_{TT_k}, the frame size l_{TT_k} and the period p_{TT_k} are known before entering the network. In the worst-case arrival scenario, there is a frame transmission at the beginning of each period. Therefore, flow τ_{TT_k} can be constrained by a leaky bucket arrival curve in the output port h_0 of the source ES, as follows

$$\alpha^{h_0}_{TT_k}(t) = \sigma^{h_0}_{TT_k} + \rho^{h_0}_{TT_k}. \quad (24)$$

where $\sigma^{h_0}_{TT_k} = l_{TT_k}$, $\rho^{h_0}_{TT_k} = l_{TT_k}/p_{TT_k}$. In the subsequent output port h along the dataflow route of τ_{TT_k}, the input arrival curve can be calculated from the input arrival curve in the previous output port h',

$$\alpha^{h}_{TT_k}(t) = \alpha^{h'}_{TT_k}(t + D^{h'}_{TT_k}). \quad (25)$$

where $D^{h'}_{TT_k}$ is the worst-case queuing latency of τ_{TT_k} in h', which can be calculated from the formula (27) below.

Moreover, since critical flows are asynchronous, all with the same priority can be backlogged in the same queue and compete for the output port h. Hence, the aggregate arrival curve $\alpha^h_{P_m}(t)$ of competing critical flows $\tau_{TT_k} \in \tau^h_{P_m}$ is the sum of their respective arrival curves

$$\alpha^h_{P_m}(t) = \sum_{\tau_{TT_k} \in \tau_{P_m}} \alpha^h_{TT_k}(t). \quad (26)$$

According to network calculus theory, the upper bound latency for the flow $\tau_{TT_k} \in \tau^h_{P_m}$ in the output port h is the maximum horizontal latency deviation between the aggregate arrival curve $\alpha^h_{P_m}(t)$ and the service curve $\beta^h_{P_m}(t)$ for the P_m traffic, namely

$$D^h_{TT_k} = h(\alpha^h_{P_m}(t), \beta^h_{P_m}(t)). \quad (27)$$

By disseminating the computation of latency bounds along the dataflow route of τ_{TT_k}, the worst-case end-to-end latency for the critical flow τ_{TT_k} is obtained by the sum of latency bounds from its source to its destination nodes as follows

$$D_{TT_k} = \sum_{h \in dr_{TT_k}, h \ne DES} D^h_{TT_k} + (|h| - 1) \cdot d_{tech}, \quad (28)$$

where DES is the destination end system of dr_{TT_k}, $|h|$ is the number of nodes along the routing dr_{TT_k}, and d_{tech} is the constant maximum latency delay of the switching fabric.

E. NOTATIONS

It is assumed that there are n ($n \in [1, 8]$) queues for critical traffic. We also denote:

- P_m, $m \in [1, n]$, priority levels for critical traffic;
- P_{m^+}, $m^+ \in [m+1, n]$, lower priority level than P_m;
- P_{m^-}, $m^- \in [1, m-1]$, higher priority level than P_m;
- Q_{P_m}, the queue used for storing the priority P_m traffic;
- G_{P_m}, the gate for the queue Q_{P_m};
- T_{P_m}, the open-close cycle of the gate G_{P_m};
- L_{P_m}, the length of open window in an open-close cycle T_{P_m};
- $\overline{L}_{P_m}$, the guaranteed time slot in an open window;
- S_{P_m}, the maximum waiting time for the P_m traffic at its beginning backlog period;

- T_{GCL}, the hyperperiod, which is the Least Common Multiple (LCM) of open-close cycles for all the priority queues of the critical traffic;
- N_{P_m}, the limited number of length cases $\overline{L}_{P_m}$ of guaranteed time slot for P_m traffic;
- $o^{j,i}_{P_m}, j \in [i+1, i+N_{P_m}-1]$, the relative offset, which is the time interval between the starting time of the ith and jth guaranteed time slots for P_m traffic, by taking the ith guaranteed time slot as the reference;
- $\overline{L}^{i}_{P_m}, i \in [1, N_{P_m}]$, the length of the ith guaranteed time slot for P_m traffic;
- $\beta^{i}_{P_m}(t)$, a possible service curve for the critical traffic of priority P_m by considering the guaranteed time slot $\overline{L}^{i}_{P_m}$ as the benchmark;
- $S^{i}_{P_m}$, the maximum waiting time for the P_m traffic at its beginning backlog period by considering the guaranteed time slot $\overline{L}^{i}_{P_m}$ as the benchmark;
- $\beta_{P_m}(t)$, the service curve for P_m traffic;
- $w^{i}_{P_m}$, the ith open window;
- $t^{o,i}_{P_m}$, the beginning time of the open window $w^{i}_{P_m}$;
- $t^{c,i}_{P_m}$, the end time of the open window $w^{i}_{P_m}$;
- $d^{np,i}_{P_{m^+}}$, the worst-case non-preemption latency for the P_m traffic in the open window $w^{i}_{P_m}$ caused by a lower priority frame from a queue $Q_{P_{m^+}}$;
- $d^{np,i}_{L}$, the worst-case non-preemption latency for the P_m traffic in the open window $w^{i}_{P_m}$;
- $t^{np,i}_{L}$, the starting time of guaranteed time slot $\overline{L}^{i}_{P_m}$ for P_m traffic in $w^{i}_{P_m}$ by just considering the lower priority traffic;
- $l^{max}_{P_m}$, the maximum frame size of all the flows with priority P_m;
- $l^{max}_{P_{m^+}}$, the maximum frame size of lower priority flows of which open windows are overlapping with $w^{i}_{P_m}$;
- $d^{gb}_{P_m}$, the maximum guard band latency;
- $t^{gb,i}_{P_m}$, the end time of guaranteed time slot $\overline{L}^{i}_{P_m}$ for P_m traffic in $w^{i}_{P_m}$ by just considering the lookahead mechanism;
- $t^{c,i}_{P_{m^-}}$, the earliest time no earlier than $t^{o,i}_{P_m}$ with the closed state of the gate $G_{P_{m^-}}$;
- $t^{o,i}_{P_{m^-}}$, the latest time no later than $t^{c,i}_{P_m}$ with the closed state of the gate $G_{P_{m^-}}$;
- $t^{B,i}_{H}$, the starting time of guaranteed time slot $\overline{L}^{i}_{P_m}$ for P_m traffic in $w^{i}_{P_m}$ by just considering the higher priority traffic;
- $t^{E,i}_{H}$, the end time of guaranteed time slot $\overline{L}^{i}_{P_m}$ for P_m traffic in $w^{i}_{P_m}$ by just considering the higher priority traffic;
- $t^{B,i}_{P_m}$, the starting time of guaranteed time slot $\overline{L}^{i}_{P_m}$ for P_m traffic in $w^{i}_{P_m}$;
- $t^{E,i}_{P_m}$, the end time of guaranteed time slot $\overline{L}^{i}_{P_m}$ for P_m traffic in $w^{i}_{P_m}$;
- $t^{o,i-1}_{P_{m^+}}$, the latest time no later than $t^{E,i-1}_{P_m}$ with the closed state of the gate $G_{P_{m^+}}$;
- $d^{np,0}_{L}$, the worst-case non-preemption latency at the beginning of the backlog period.

VI. EXPERIMENTAL RESULTS

In this section, we first analyze the WCDs using our proposed method on different smaller synthetic test cases, in order to show the influence of different overlapping cases of GCL transmission windows and different priority assignments of flows on the WCDs. Further, we also perform experiments on a larger real-world use-case adapted from the Orion Crew Exploration Vehicle (CEV) described in [9] in order to show the scalability of our analysis.

A. SYNTHETIC TEST CASES

In this section, we take one TT flow as the reference target to observe the factors that affect its WCD. Our evaluation focuses on a tree topology of 6 ESes and 2 SWs, connected via physical links with rates of 1Gb/s, shown in Fig. 12. As mentioned, we assume that the GCLs for each priority queue of the output ports are given as an input. The test case has 13 critical TT flows with different priorities ranging from 1 to 7, running on 10 routes (some TT flows share the same route). We take τ_{TT_1} as the target flow.

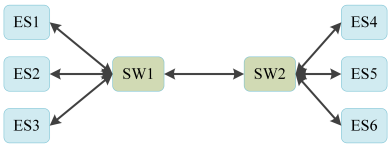


FIGURE 12. The topology of the synthetic small test cases.

For the first three experiments, τ_{TT_1} is configured with the frame size of 400 bytes, a period of 250 μs and an assigned priority of 2. Its routing is shown in the first column of Table 1, consisting of an ordered sequence of dataflow links $[[ES_2, SW_1], [SW_1, SW_2], [SW_2, ES_6]]$, which is a directed communication connection from v_a to v_b, where v_a and v_b can be ESes or SWs. Table 1 also gives the GCLs information for τ_{TT_1} along its route, including open-close cycles and opening and closing times of transmission windows configured in the GCL for each priority queue. In addition, the maximum frame sizes for each priority queue are shown in the last column.

In the first experiment, we are interested in evaluating how the overlapping variation influences the WCD of τ_{TT_1}. Therefore, we keep the overlapping scenarios of all the priority queues unchanged except for the priority 2 queue for τ_{TT_1}, as shown in Table 2. In Case 1, as the open windows for the queue of priority 2 are not overlapped with open windows for other priority queues, TT1 has the minimum WCD compared with other three cases. In Case 2, the open windows for the queue of priority 2 are overlapped with open windows from a lower priority queue in each output port along the path of

TABLE 1. GCLs along the routing of τ_{TT_1}.

Link	Priority	Open-Close (μs)	Period (μs)	Max Frame Size (B)
$[ES_2, SW_1]$	7	[70, 90]	250	400
	5	[85, 105]	250	400
	3	[40, 60]	250	400
	2	[95, 115]	250	400
	1	[55, 75]	250	400
$[SW_1, SW_2]$	7	[70, 90]	250	400
	6	[115, 135]	250	400
	5	[130, 150]	250	400
	4	[145, 165]	250	400
	3	[85, 105]	250	400
	2	[155, 175]	250	400
	1	[100, 120]	250	400
$[SW_2, ES_6]$	5	[170, 190]	250	400
	2	[180, 200]	250	400
	1	[140, 160]	250	400

TABLE 2. WCD of τ_{TT_1} under different overlapping scenarios configured in the GCL.

Case	Priority	Link	Open-Close (μs)	WCD of TT1 (μs)
1	2	$[ES_2, SW_1]$	[105, 125]	1036.6
		$[SW_1, SW_2]$	[165, 185]	
		$[SW_2, ES_6]$	[190, 210]	
2	2	$[ES_2, SW_1]$	[95, 115]	1287.8
		$[SW_1, SW_2]$	[155, 175]	
		$[SW_2, ES_6]$	[180, 200]	
3	2	$[ES_2, SW_1]$	[80, 100]	1173.0
		$[SW_1, SW_2]$	[140, 160]	
		$[SW_2, ES_6]$	[160, 180]	
4	2	$[ES_2, SW_1]$	[65, 85]	2309.6
		$[SW_1, SW_2]$	[110, 130]	
		$[SW_2, ES_6]$	[150, 170]	

τ_{TT_1}, i.e., $[[ES_2, SW_1], [SW_1, SW_2], [SW_2, ES_6]]$. Moreover, the open time of the gate for the lower priority queue starts earlier than the open time of the gate for the queue with priority 2. For example, in the output port $[ES_2, SW_1]$, the open duration in an open-close cycle for queues with priority 5 and 2 are respectively [85 μs, 105 μs] (Table 1) and [95 μs, 115 μs] (Table 2). Therefore, the guaranteed service for the traffic with priority 2 is influenced by both the lower priority traffic as well as by the guard band blocking, causing higher WCD of τ_{TT_1} than in Case 1. Case 3 is similar to Case 2 in that the open windows for priority 2 are overlapped with lower priority open windows. However, in the output port $[SW_2, ES_6]$, the open time 170 μs of the gate for priority 5 starts later than the open time 160 μs of the gate for priority 2, thus the guaranteed service for traffic with priority 2 in the output port $[SW_2, ES_6]$ is only related to guard band blocking. Hence, the WCD of τ_{TT_1} is smaller than the result from Case 2. For Case 4, open windows for the queue with priority 2 are overlapped with open windows for the higher priority queue in each output port along the path of τ_{TT_1}. As the guaranteed service for traffic with priority 2 is related to the whole overlapping of open windows of the higher priority queue, the WCD of τ_{TT_1} is larger than in the other three cases.

TABLE 3. WCDs of τ_{TT_1} under different lengths of open windows of the GCL for the queue with priority 2.

Case	Priority	Link	Open-Close (μs)	WCD of TT1 (μs)
1	2	$[ES_2, SW_1]$	[95, 110]	1797.1
		$[SW_1, SW_2]$	[155, 170]	
		$[SW_2, ES_6]$	[180, 195]	
2	2	$[ES_2, SW_1]$	[95, 115]	1287.8
		$[SW_1, SW_2]$	[155, 175]	
		$[SW_2, ES_6]$	[180, 200]	
3	2	$[ES_2, SW_1]$	[95, 125]	744.7
		$[SW_1, SW_2]$	[155, 185]	
		$[SW_2, ES_6]$	[180, 210]	

TABLE 4. WCDs of τ_{TT_1} under different open-close cycles of the GCL for the queue with priority 2.ss.

Case	Priority	Link	Open-Close (μs)	Period (μs)	WCD of TT1 (μs)
1	2	$[ES_2, SW_1]$	[95, 115]	350	2527.8
2	2	$[SW_1, SW_2]$	[155, 175]	250	1287.8
3	2	$[SW_2, ES_6]$	[180, 200]	100	474.0

TABLE 5. WCDs of τ_{TT_1} under different priority assigned.

Case	Priority	WCD of TT1 (μs)
1	1	1177.9
2	2	1287.8
3	5	2014.0

In the second experiment, we change the length of open duration in an open-close cycle for the queue with priority 2 to show the effects on the WCD of τ_{TT_1}. As shown in Table 3, the open times of the gate for the queue with priority 2 from Case 1 to Case 3 are the same in the same output port. We use different closing times of the gate to create the variable length of open duration of GCL for the queue with priority 2. As expected and can be seen from the last column in Table 3, the WCD of τ_{TT_1} decreases with the increasing length of the corresponding open windows of the GCL.

For the third experiment, we show the influence of the variations in open-close cycles of the GCL on the WCD of τ_{TT_1}. The overlapping situations and the lengths of open windows for each case are fixed, as shown in Table 4. However, we change the open-close cycle of the GCL for the queue with priority 2. As expected, with a reduced open-close cycle, the guaranteed service for τ_{TT_1} with priority 2 is increased, thus causing a decrease in the WCD of τ_{TT_1}.

For the last synthetic test case, we are interested in showing the influence of different priorities assigned to τ_{TT_1} on its WCD. Here, the GCL is still used from the configuration described in Table 1, and the variation of priority assignments for τ_{TT_1} is shown in Table 5. As expected, the higher the priority assigned to τ_{TT_1} is, the smaller its WCD becomes.

B. EVALUATION ON A LARGER REALISTIC TEST CASE

For the next set of experiments where we want to show the scalability of our analysis, we use a larger real-world test case,

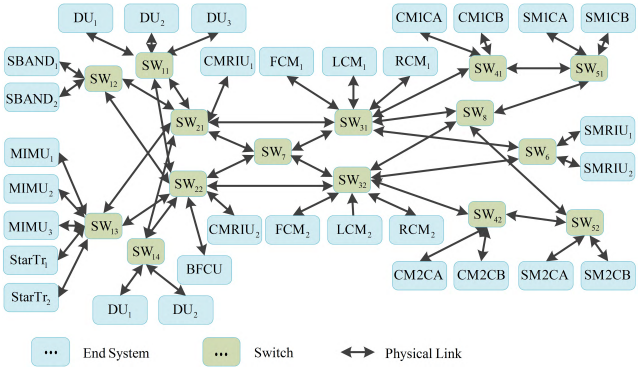


FIGURE 13. Network topology of the Orion CEV [9].

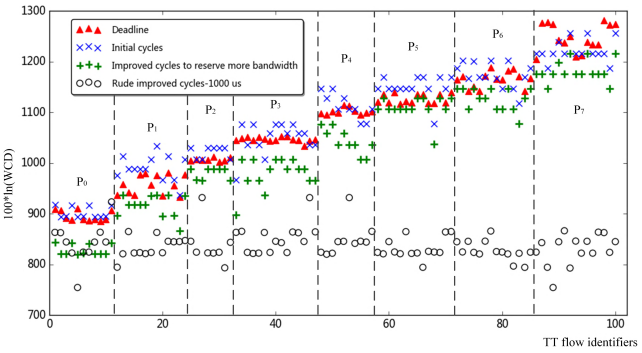


FIGURE 14. Compared WCDs under different cycle of GCL of priority queues.

TABLE 6. GCL for the priority queue in an output port.

Priority	Open-Close (μs)	Period (μs)	Num of TT flows
0	[0, 100]	2000	11
1	[80, 180]	4000	13
2	[160, 260]	6000	8
3	[240, 340]	8000	15
4	[320, 420]	16000	10
5	[400, 500]	24000	14
6	[480, 580]	24000	14
7	[560, 660]	48000	15

adapted from the Orion Crew Exploration Vehicle (CEV) by using the same topologies and TT flows as described in [9].

CEV has a topology consisting of 31 ESes, 15 SWs and 188 routes connected by dataflow links transmitting at 1Gb/s, as shown in Fig. 13. There are 100 TT flows (including multicast flows) in the CEV. The deadlines for TT flows are presented in Fig. 14, which are marked with red triangles. In order to clearly show WCDs with large numerical gaps, the y-axis in Fig. 14 uses a logarithmic scale with $100 \cdot ln(\text{WCD})$. In addition, WCDs of TT flows with different priorities are separated with a vertical dotted line in Fig. 14. The x-axis represents the identifiers of TT flows. Table 6 describes the manually designed GCLs for each priority queue in an output port of a node, as well as the number of TT flows with the corresponding priority in the network.

The logarithm deformation of WCDs for each TT flows following the designed GCL in Table 6 is shown with blue "x" symbols in Fig. 14. As we can see, most of the resulting WCDs exceed the respective required deadlines of the TT flows. In accordance with the discussion in Sect. VI-A, we can reduce the WCDs by decreasing the open-close cycles of the GCL for each priority queue.

First, we try to reduce the open-close cycles of GCLs for all the priority queues to 1000 μs. In this case we obtain the WCDs for each TT flows shown with circles in Fig. 14. As can be seen, the results are all below the required deadlines (red triangles) of the TT flows. However, such a configuration leads to wasted bandwidth due to overconstraining of TT flows with lower priority, which is not necessary.

Therefore, in the following, we change the open-close cycles of GCLs for priority [0 – 7] respectively to [1000, 2000, 4000, 4000, 8000, 16000, 16000, 32000]. In this case, the WCDs for each TT flow are marked with green crosses in Fig. 14. All the resulting WCDs are below the required respective deadlines thus satisfying the timeliness requirements. Moreover, due to the differences between the results and the corresponding deadlines additional bandwidth is available for other kinds of traffic, such as Audio-Video-Bridging (AVB) or Best-Effort (BE) traffic.

Obviously, we can also improve the WCDs of TT flows by synthetically changing the overlapping situations of open windows or the length of open durations of the GCLs for different priority queues, or by assigning a different priority to TT flows as discussed in Sect. VI-A. Therefore, the experiments in this section not only show the scalability of our analysis method, but also demonstrate the potential of our method to guide the GCL synthesis by using optimization algorithms, acting as a feedback loop that provides the optimization with latency results which can be used to guide the search.

VII. CONCLUSION

Distributed applications in domains like industrial automation, automotive, and aerospace, implemented using Ethernet-based network technologies such as Time-Sensitive Networks, require certification evidence that proves the correct real-time behavior of critical communication flows. Motivated by this, we derive a analysis based on network calculus of the worst-case latency for critical communication flows that takes into account both traffic priority assignment as well as the overlapping of transmission windows configured in the GCLs. We validate our approach in terms of scalability and the effect of GCL overlapping characteristics on individual flows using experiments on both synthetic and real-world test cases.

The presented analysis can be used both for analysis and certification of critical communication in TSN networks as well as for driving GCL synthesis optimization algorithms that require a worst-case delay analysis in order to compute the open and close events for the global schedule.

REFERENCES

[1] *Time-Sensitive Networking Task Group*, IEEE Standards 802.1, Institute of Electrical and Electronics Engineers, 2016. Accessed: Jul. 6, 2016. [Online]. Available: http://www.ieee802.org/1/pages/tsn.html

[2] *802.1Qbv—Enhancements for Scheduled Traffic*, IEEE Standards 802.1, Draft 3.1, Institute of Electrical and Electronics Engineers, 2016. [Online]. Available: http://www.ieee802.org/1/pages/802.1bv.html
[3] S. S. Craciunas, R. S. Oliver, M. Chmelik, and W. Steiner, "Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks," in *Proc. 24th Int. Conf. Real-Time Netw. Syst. (RTNS)*, Oct. 2016, pp. 183–192.
[4] 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-Phys. Syst., Theory Appl.*, vol. 1, no. 1, pp. 86–94, 2016.
[5] R. S. Oliver, S. S. Craciunas, and W. Steiner, "IEEE 802.1Qbv gate control list synthesis using array theory encoding," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Apr. 2018, pp. 1–12.
[6] F. Frances, C. Fraboul, and J. Grieu, "Using network calculus to optimize the AFDX network," in *Proc. Embeeded Real Time Softw. Syst. (ERTS)*, Jan. 2006, pp. 1–8.
[7] J. Diemer, D. Thiele, and R. Ernst, "Formal worst-case timing analysis of Ethernet topologies with strict-priority and AVB switching," in *Proc. Int. Symp. Ind. Embedded Syst. (SIES)*, Jun. 2012, pp. 1–10.
[8] H. Bauer, J. L. Scharbarg, and C. Fraboul, "Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach," *IEEE Trans. Ind. Informat.*, vol. 6, no. 4, pp. 521–533, Nov. 2010.
[9] L. Zhao, P. Pop, Q. Li, J. Chen, and H. Xiong, "Timing analysis of rate-constrained traffic in TTEthernet using network calculus," *Real-Time Syst.*, vol. 52, no. 2, pp. 254–287, 2017.
[10] L. X. Zhao, H. G. Xiong, Z. Zheng, and Q. Li, "Improving worst-case latency analysis for rate-constrained traffic in the time-triggered Ethernet network," *IEEE Commun. Lett.*, vol. 18, no. 11, pp. 1927–1930, Nov. 2014.
[11] M. Boyer, H. Daigmorte, N. Navet, and J. Migge, "Performance impact of the interactions between time-triggered and rate-constrained transmissions in TTEthernet," in *Proc. Eur. Congr. Embedded Real Time Softw. Syst.*, 2016.
[12] E. Wandeler and L. Thiele, "Optimal TDMA time slot and cycle length allocation for hard real-time systems," in *Proc. Asia South Pacific Conf. Design Automat.*, Jan. 2006, pp. 479–484.
[13] D. D. Khanh and A. Mifdaoui, "Timing analysis of TDMA-based networks using network calculus and integer linear programming," in *Proc. Modelling, Anal. Simulation Comput. Telecommun. Syst.*, Sep. 2014, pp. 21–30.
[14] L. Zhao, P. Pop, Z. Zheng, and Q. Li, "Timing analysis of AVB traffic in TSN networks using network calculus," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Jun. 2018, pp. 1–8.
[15] D. Thiele, R. Ernst, and J. Diemer, "Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers," in *Proc. IEEE Veh. Netw. Conf.*, Dec. 2015, pp. 251–258.
[16] R. L. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.
[17] J. Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer, 2001.
[18] C. S. Chang, *Performance Guarantees in Communication Networks*. Springer, 2000.

LUXI ZHAO received the Ph.D. degree in communication and information system from Beihang University, Beijing, China, in 2017. She has been holding a post-doctoral position at the Technical University of Denmark since 2017. Her main research interest concerns worst-case analysis, performance evaluation, and the configuration of real-time and safety-critical real-time networks.

PAUL POP received the Ph.D. degree in computer systems from Linköping University in 2003. He has been an Associate Professor at DTU Compute, Technical University of Denmark, where he has been a Professor of cyber-physical systems since 2016. His research is focused on developing methods and tools for the analysis and optimization of dependable embedded 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 has also received the EDAA Outstanding Dissertations Award (Co-Supervisor) in 2011. His research has been highlighted as The Most Influential Papers of 10 Years DATE. He is the Chairman of the IEEE Danish Chapter on Embedded Systems. He is the Director of the DTU's IoT Research Center and has coordinated the Danish National InfinIT Safety-Critical Systems Interest Group.

SILVIU S. CRACIUNAS received the Ph.D. (Dr.techn.) degree from the University of Salzburg, Austria, in 2010, and the Dipl.Ing. degree from the Politehnica University of Timisoara, Romania, in 2005. He was a Post-Doctoral Researcher with the Embedded Software and Systems Research Center, University of Salzburg. He joined TTTech Computertechnik AG, Vienna, in 2011, where he is currently a Core Architect, working on scheduling algorithms for time triggered networked systems and real-time operating systems. His research interests include real-time and safety-critical embedded systems, real-time scheduling algorithms, power-aware computing, cyber-physical systems, and deterministic real-time networks.