Determining a tight worst-case delay of switched Ethernet network in IEC 61850 architectures

Théo Docquier, Ye-Qiong Song, Vincent Chevrier, Ludovic Pontnau, Abdelaziz Ahmed-Nacer

Determining a tight worst-case delay of switched Ethernet network in IEC 61850 architectures

Théo Docquier
University of Lorraine, CNRS, LORIA, SCLE SFE
Villers-lès-Nancy, France
theo.docquier@loria.fr

Ye-Qiong Song
University of Lorraine, CNRS, LORIA
Villers-lès-Nancy, France
ye-qiong.song@loria.fr

Vincent Chevrier
University of Lorraine, CNRS, LORIA
Villers-lès-Nancy, France
vincent.chevrier@loria.fr

Ludovic Pontnau
SCLE SFE
Toulouse, France
ludovic.pontnau@scle.fr

Abdelaziz Ahmed-Nacer
SCLE SFE
Toulouse, France
abdelaziz.ahmed-nacer@scle.fr

***Abstract*—IEC 61850 has become the reference standard for Substation Automation Systems (SAS) in smart power grids. Switched Ethernet is used for machine to machine communication within SAS. In order to meet stringent real-time constraints, the IEC 61850 application layer protocols can be mapped into different IEEE802.1Q priorities according to their real-time constraints and application criticality. However, the delay evaluation to guarantee real-time requirements can be difficult to perform, especially for lower priority but still real-time constrained traffic. In fact, most existing end-to-end worst-case delay analyses provide upper-bounds, leading to some pessimism and consequently network resource over-provision. In this paper, we present a new method for determining a tight worst-case delay. This method is based on the study of flow characteristics from a given network path. As a flow is interfered by other concurrent flows on its path, their relative offsets with the considered flow greatly impact on its delay. Studying all combinations to find the actual worst-case delay results in high complexity. We show that this complexity can be reduced by only analysing local worst-case delay at each switch in stead of the whole path where the change at each switch would need re-analysing the already analysed switches. An algorithm is also proposed to perform the analysis. An illustrating example shows that our method can reduce the pessimism as it provides the tight worst-case delay instead of the upper-bound of the worst-case delay.**

***Index Terms*—switched Ethernet, worst-case delay analysis, real-time, Strict-Priority, IEC 61850, smart grid.**

I. Introduction

Nowadays, power utilities need to provide high quality of energy distribution to companies and customers. It becomes challenging with the progressive introduction of micro-grids, renewable sources and new appliances such as electric vehicles. To guarantee a reliable power distribution, utilities rely on substations to real-timely control and protect the electrical equipment [1], [2]. A Substation Automation System (SAS) is composed of equipment scattered through the station and dealing with specific tasks (protection, monitoring, control, ...) in a distributed way. These equipment are connected to Intelligent Electronic Devices (IED) which constitute the interface between the physical system and the network. The IEC 61850 standard has been developed in order to guarantee the interoperability of equipment interconnected by networks. It standardizes the use of Ethernet as the main networking technology and includes several application protocols to meet the different SAS application requirements. There are three main application layer protocols: Sampled Measured Value (SMV), Generic Object Oriented Substation Event (GOOSE) and Manufacturing Message Specification (MMS). All IEC 61850 protocols have to meet different time constraints, as specified in [3]. Therefore, utilities have to meet real-time challenges.

An illustrative example of a Substation associated to its communication network architecture is given in Figure 1. In this example, the power comes from either a power source or another substation (*Input power line* in Figure 1) and is transmitted to either other substations or directly to customers (*Output power line* in Figure 1). IEDs are physically connected to electrical equipments like sensors (e.g., the *current transformer* in Figure 1) or actuators (e.g., the *circuit breaker* in Figure 1), in order to real-time collecting data from the physical process or to transmit a control order. Information between IEDs is transmitted through the communication network to one or multiple destinations in order to carry out functional tasks. Messages can be transmitted using IEC 61850 protocols as well as other "classic" protocols (e.g., FTP for file transfer, SNMP for network management, NTP/PTP for time synchronization...).

The main issue is to ensure that all messages with real-time requirements will be delivered on time in a switched Ethernet network context, without overprovisioning network resources. One solution is to provide a mapping between messages belonging to IEC 61850 protocols and the Ethernet 802.1Q priority levels. GOOSE messages are mapped with high priority level as they are used for high-

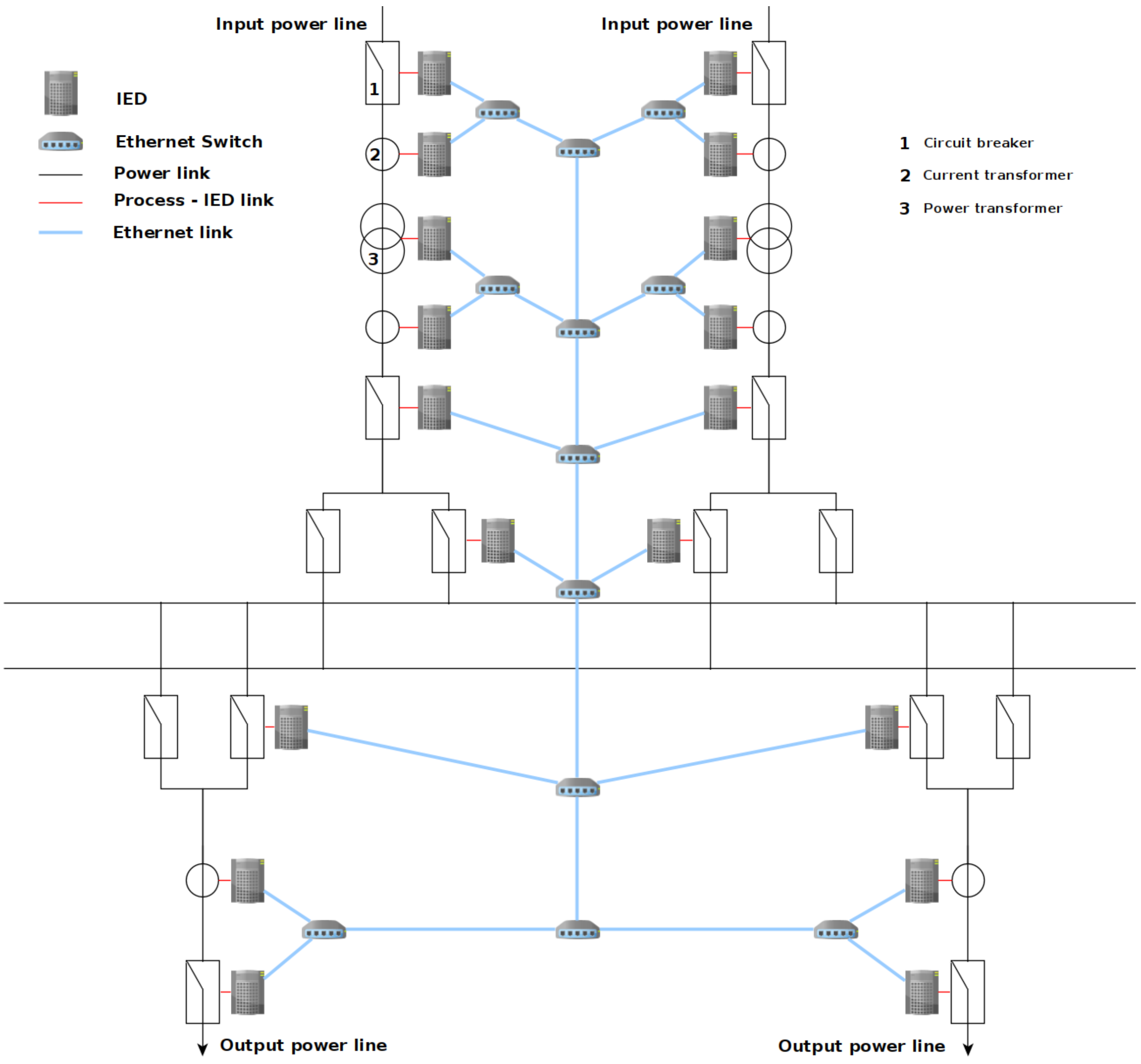

Fig. 1. Simple example of a substation and its associate communication network

critical operations (applications belonging to TT5-6 classes in Table I). MMS messages are mapped with a lower priority level than GOOSE since they fulfill operations with less stringent requirements (operations belonging to TT0 to TT4 classes in Table I). SMV messages are considered as critical as GOOSE and have traditionally the same priority (operations belonging to TT6 class in Table I). We then have to apply a scheduling policy (e.g., Strict Priority, Weighted Round Robin, ...), based on the related priorities. In IEC 61850 context, Strict Priority is often used as the scheduling policy. Even though the mapping gives more resources to high-priority messages, the end-to-end worst-case delay could be difficult to determine, especially for low-priority messages.

Transfer time class	Transfer time [ms]	Application examples: Transfer of
TT0	>1000	Files, events, log, contents
TT1	1000	Events, alarms
TT2	500	Operator commands
TT3	100	Slow automatic interactions
TT4	20	Fast automatic interactions
TT5	10	Releases, status changes
TT6	3	Trips, blockings

TABLE I
PERFORMANCE REQUIREMENTS FOR DIFFERENT IEC 61850 APPLICATIONS [3]

To address this issue, several approaches have been proposed. Simulation is a common method used both for its simplicity, scalability and the possibility to obtain a more realistic view of the results. The worst-case delay is however difficult to reach and it needs theoretically an infinite number of simulations to obtain it (rare event). The worst-case delay can also be obtained by exhaustively analysing all possible combinations of traffic flows in a network, using for instance model checking technique. For evaluating the worst-case end-to-end delay, as all possible network flows with all possible relative offsets between them must be examined, it is in general unscalable facing to the combinatorial explosion problem. Other approaches, such as those based on the network calculus searching for an upper bounded worst-case delay, have also been proposed. Considering maximal arrival curve and minimal service curve can indeed simplify the analysis and ensure a safe upper-bound on delay. Unfortunately, there is a trade-off between the complexity of the used approach and the pessimism obtained for the worst-case delay bound. This pessimism leads to network overprovisioning, especially with high number of switch hops.

The main contribution of this paper is to propose, for any network's path that a flow goes through, an iterative approach to determine the Worst-Case Delay (WCD) by eliminating all impossible scenarii. As only possible flow offsets are considered, the obtained delay is thus tight by design. Our method is based on the independent local worst-case analysis that considerably reduces the analysis complexity comparing to an exhaustive search of global flow combinations. We first try to find out what is the maximal worst-case delay for a node (end system and switch) and whether it is reachable or not with different flow offsets at their source nodes. If this maximal worst-case delay is not possible, we determine another delay, necessarily lower than the maximal one but being the highest possible. In order to avoid a complex analysis, we work at a flow level instead of a frame level. At the end, we want to obtain a tight end-to-end WCD for all IEC 61850 application protocols, i.e., for protocols using different Ethernet priorities. The determination of the tight WCD allows either to optimize the network architecture or anticipate the architecture scalability.

The rest of this paper is organized as follows: Section II gives more detail about the related work and shows through a motivating example that we can effectively improve the worst-case delay obtained by the Compositional Performance Analysis (CPA). Section III gives elements for the problem formulation. Section IV specifies the method to obtain the tight WCD, as well as the associated iterative algorithm. Section V presents a numerical example illustrating how our proposition can be applied. Finally, Section VI concludes and points out our future work.

II. Related work and motivating example

A. Related works

Different with delay evaluation in general-purpose networks where we do not have stringent deadline (real-time) constraints and queuing theory can be applied to obtain stochastic performance indicators, real-time networking focuses on the worst-case delay evaluation for providing deterministic deadline guarantee to real-time applications. Switched Ethernet and its recent evolution to TSN (time sensitive networking) have attracted many research work.

Searching for the WCD through a switched Ethernet network has been extensively studied over the years. The difficulty of such a task lies in the complexity of the architecture, i.e., the network topology, the number of end-devices scattered through the network, the flow characteristics generated by the source nodes and relative offsets between them, etc... Basically each flow is characterized by the size of its data packet (i.e., Ethernet frame), its generating period (or minimal inter-arrival interval) and its priority level together with its source node, the path it passes through, and its destination node(s). The fundamental issue is to find the worst-case combinations of the flows in order to determine the WCD for each flow of a given priority. In general we consider its generating period as its deadline to be met by the network, that means a frame must be transmitted to its destination before the next one is generated.

The real-time task scheduling theory has been extended e.g., by Tindell and Clark [4] and further by Pop et al. [5], taking into account distributed hard-real time systems with static priorities to obtain a worst-case delay. Unfortunately, these approaches become difficult to handle as the network architecture grows in complexity. An exhaustive investigation of all possible flow offset combinations is performed by Charara et al. [6] using the model checking approach. The work is later improved by Adnan et al. [7], by reducing the space of possible combinations to analyze, giving a tighter worst-case response-time for more complex architectures. Nevertheless, the resulting combinatorial explosion of such approach, as the network size grows, does not make their use appropriate to industrial size use cases.

To reduce the complexity issue, other approaches have been proposed. Georges et al. [8] use Network Calculus (NC) to switched Ethernet network with static priorities, giving a method to determine the worst-case delay. A later work by Georg et al. [9] uses NC in the IEC 61850 context. However, the intrinsic pessimism of NC does not permit to obtain the tight WCD. In fact, for reducing the analysis complexity, NC considers a maximal arrival curve and minimal service curve, leading to a safe, but over-estimated delay bound. The trajectory Approach with static priorities has been applied with success in [10], with a guaranteed delay bound for AFDX architectures (i.e., switched Etnernet for avionic networks). However, pessimism for the delay bound still remains, its origin being deeper investigated in [11]. Diemer et al. [12] study an Ethernet with Strict-Priority case using the Compositional Performance Analysis (CPA). This approach uses scheduling analysis at lower granularity level, i.e., on subpart of the system called component, drastically reducing the complexity. However, this complexity reduction leads to pessimism as for the obtained WCD bound.

We developed in a previous work [13] a tool to automatically generate a network architecture from IEC 61850 network configuration files. This work is based on the one made by Léon et al. [14], where an IEC 61850 model was proposed. Even though we can run simulations with scenarios that make it easier to get the worst case delay, the latter may actually never be reached since the worst-case is rare event during simulations. To make up for this issue, we propose an analysis to obtain the tight worst-case delay for the GOOSE protocol in [15]. However, this work is performed in a Time Sensitive Networking (TSN) context and does not cover other protocols than GOOSE.

B. Motivating example

We take a simple network architecture proposed by Rox and Ernst [16] as shown in Figure 2, where CPA is used to address the worst-case delay issue.

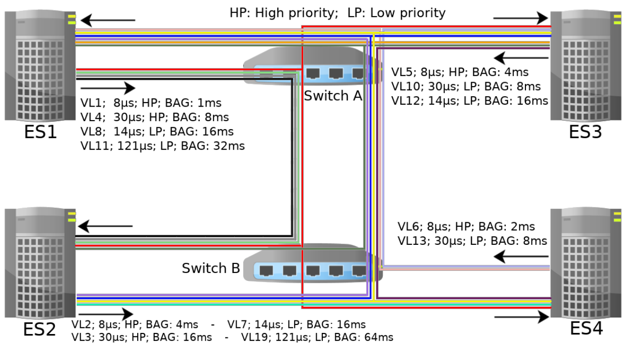
HP: High priority; LP: Low priority
ES1
VL1; 8µs; HP; BAG: 1ms
VL4; 30µs; HP; BAG: 8ms
VL8; 14µs; LP; BAG: 16ms
VL11; 121µs; LP; BAG: 32ms
Switch A
VL5; 8µs; HP; BAG: 4ms
VL10; 30µs; LP; BAG: 8ms
VL12; 14µs; LP; BAG: 16ms
ES3
VL6; 8µs; HP; BAG: 2ms
VL13; 30µs; LP; BAG: 8ms
Switch B
ES2
VL2; 8µs; HP; BAG: 4ms - VL7; 14µs; LP; BAG: 16ms
VL3; 30µs; HP; BAG: 16ms - VL19; 121µs; LP; BAG: 64ms
ES4

Fig. 2. Architecture example taken from [16]

The flows are called VL (Virtual links according to the terminology of AFDX). Each of them is characterized by its source ES (End System), destination ES, the frame transmission time T_{tr}, the priority and the frame minimal inter-arrival interval BAG (Bandwidth Allocation Gap).

We consider a given set of flows VL as shown in Figure 2, and focus on the WCD of flow VL11 going from ES1 to ES2 through switch A and B. The worst-case delay computed in [16] is 481µs. However, by counting all possible sources of delay, we obtain 459µs. Sources of delays are described as follows:

- *WCD at node ES1*: all other VL frames compete with VL11 frames to be transmitted. This gives: T_{tr}(VL1) + T_{tr}(VL4) + T_{tr}(VL8) = 8 + 30 + 14 = 52µs;

Abbreviation	Description
G	Graph of a Network
Fc	Concurrent Flow
Fm	Main Flow
Fo	Outgoing Flow
HP	High Priority
LP	Low Priority
MF	Main Frame
P	Path on G
$PLWCD$	Possible Local Worst-Case Delay
SP	Same Priority
$TLWCD$	Theoretical Local Worst-Case Delay
v	Vertex
WCD	Worst-Case delay

TABLE II
LIST OF ABBREVIATIONS

- *WCD at Switch A*: all frames coming from ES3 and going to either ES2 or ES4 compete with frames coming from ES1. This gives: T_{tr}(VL10) + T_{tr}(VL12) = 30 + 14 = 44µs;
- *WCD at Switch B*: all frames coming from ES4 and going to ES2 compete with frames coming from switch A. As there is no VL corresponding to this case, no delay can block the VL11 frame;
- *Transmission time*: as VL11 had three hops to pass to reach ES2, we have to count its transmission time three times. This gives: T_{tr}(VL11) $\times$ 3 = 121 $\times$ 3 = 363µs.

Adding up all delays, we have: 52+44+363= 459µs, which is less than the delay of 481µs obtained using the CPA approach given in [16].

From this simple observation, we can notice that all data with the same destination, or at least competing to access the medium on one node could be a source of extra-delay. As a consequence, we need to formalize this observation in order to determine what is the maximal delay a frame could suffer on one node. Besides, we have to determine if this maximal delay is always reachable or not according to the relative offsets of the incoming frames.

III. Problem modelling and assumptions

We detail the model of the problem in Section III-A followed by the assumptions in Section III-B. Table III summarizes notations used in the paper. Table II and Table III gives respectively abbreviations and notations used in the paper.

A. Problem description

We want to determine the Worst-Case delay (WCD) from the source to the destination for any IEC 61850 protocol message. As a message is encapsulated into an Ethernet frame, we use the term frame instead of message for the rest of the paper. We note as Main Frame (MF) the frame for which we investigate the WCD.

Notation	Description	Remarks
v_i	Vertex i on G	-
f_k	k-th frame of a flow	-
Γ_i	Set of all Fc on v_i	-
α	Same Priority (SP) frame quantity	-
β	High Priority (HP) frame quantity	-
γ	Total frame quantity	$\alpha + \beta$
γ_{q_i+1}	Total frame quantity of F_m	$\alpha_{q_i+1} + \beta_{q_i+1}$
n	Number of path vertices	-
q_i	Number of Fc on v_i	-
T_{HP_i}	HP blocking delay on v_i	Caused by HP frames
T_{SP_i}	SP blocking delay on v_i	Caused by SP frames
T_{LP_i}	Low-Priority (LP) blocking delay on v_i	Caused by LP frames
T_{HP}	sum of all HP blocking delay on P	$T_{HP} = \sum_{i=1}^{n} T_{HP_i}$
T_{SP}	sum of all SP blocking delay on P	$T_{SP} = \sum_{i=1}^{n} T_{SP_i}$
T_{LP}	sum of all LP blocking delay on P	$T_{LP} = \sum_{i=1}^{n} T_{LP_i}$
T_{tr}	Transmission delay	Include propagation and switch legacy delay

TABLE III
LIST OF NOTATIONS

The WCD is determined on a specific path P, on the network depicted as a graph $G = (V, E)$. G is a connected graph and we assume there is only one path from a vertex v_i to a vertex v_j $\forall$ v_i, $v_j \in V$. A vertex v_i is either an interconnection node (e.g., switch, router...) or an end-device (e.g., IED, server...). A frame can be stored (queued) in a vertex if it cannot be transmitted immediately. A frame which is not immediately transmitted is considered as blocked (queued). We note the succession of path vertices as $v_1, v_2, \ldots, v_n$ where n is the total number of vertices composing the path. The index i represents the i-th vertex from the first one v_1 (representing the source node) to the last one v_n (the destination node).

MF at a vertex v_i can be interfered by frames having the same destination as well as frames with other destinations. Since we are only interested in the WCD of the MF, we do not consider other output port than the one that MF passes through.

At v_i, MF may suffer from different delays during its transmission:
High-Priority blocking delay (T_{HP_i}): delay that a frame has to wait at v_i because of the transmission of the higher priority frames (denoted later as HP frames). All frames not having the highest priority may be subject to this delay.
Same-Priority blocking delay (T_{SP_i}): delay that a frame has to wait at v_i because of the already queued same priority (SP) frames' transmissions. All types of frames may be subject to this delay. This is also called the FIFO queuing delay.
Low-Priority blocking delay (T_{LP_i})**:** delay that a frame has to wait at v_i because of the transmission of frames with lower priorities. It describes the non-preemption phenomenon. Frames belonging to all protocols, excepted the lowest priority may be subject to this delay.
Transmission delay (T_{tr}): frame transmission delay. It is given by $\frac{S}{C}$, where C is the link capacity and S the frame size. It includes the propagation delay and the legacy switch latency.

T_{tr} is considered as known and constant and T_{LP_i} can be easily computed (non-preemption phenomenon). We only consider two classes of priority: the priority of MF and high priority. For example, if the MF is transmitted with the MMS protocol, all frames transmitted with either SMV or GOOSE belong to the high priority class, with BE only being able to cause one T_{LP_i} delay. T_{SP_i} and T_{HP_i} constitute the main challenge to determine.

Using this representation, the WCD for n vertices on P can be given by:

$$WCD = \sum_{i=1}^{n-1} T_{HP_i} + \sum_{i=1}^{n-1} T_{SP_i} + \sum_{i=1}^{n-1} T_{LP_i} + (n-1)T_{tr} \tag{1}$$

Where the three sums represent the delay the considered main frame may suffer on each v_i, excepted the last one (destination node). For the rest of the paper, we use the notation T_{HP}, T_{SP} and T_{LP} to designate the sum terms. The $(n-1) \times T_{tr}$ term describes the $n-1$ transmission delays to reach the destination from the source. The worst-case T_{LP} delay can be reduced to $(n-1) \times T_{LP_i}$, as at most one LP frames can block frames of higher priority at each node (excepted the last one) because of the non-preemption nature of Ethernet frames. Subsequently, following sections focus on determining T_{HP} and T_{SP}.

We formalize the notion of flow, since MF can be blocked by one or multiple frames which may be of the same or high priority. A flow F is defined as a 3-tuple [α;β;γ], with α, β, $\gamma \in \mathbb{N}$. α stands for the sum of SP frames, β the sum of HP frames and γ is the sum of α and β. A flow is said **compact** if for all frames composing the flow, we have the following relation:

$$t_k = \begin{cases} t_1, & \text{if } k = 1 \\ t_{k-1} + T_{tr}, & \text{otherwise} \end{cases} \tag{2}$$

where t_k is the arrival date of the frame f_k on v_i. An illustration is given Figure 3.

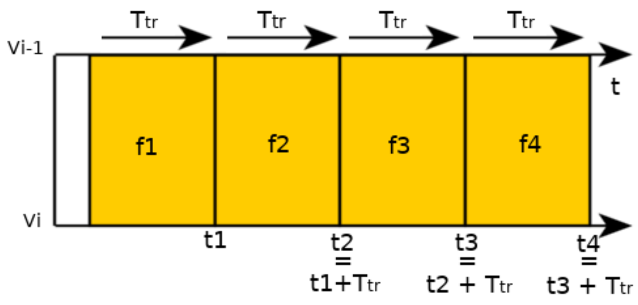

Fig. 3. Illustration of a compact flow

Each vertex v_i is described with three kinds of flows: one incoming main flow Fm, q_i incoming concurrent flows Fc and one outgoing flow Fo.

Fm is the flow containing MF. Fc is a flow that interferes with Fm to access the same output port. It is composed of frames coming from sources outside P.

Frames from Fc arrive at v_i depending on its transmission starting date, called the **frame transmission offset** in this paper, and the time to arrive from its source to v_i. Fo is the flow resulting from the competition between Fm and $q_i \times Fc$. It is given by:

$$Fo := \left[\sum_{k=1}^{q_i+1} \alpha_k ; \sum_{k=1}^{q_i+1} \beta_k ; \sum_{k=1}^{q_i+1} \gamma_k \right] \qquad (3)$$

In this paper, we assume that $\boldsymbol{Fm}$ is indexed as $\boldsymbol{q_i + 1}$. The organization of frames within Fo depends on the order in which the frames get out of v_i.

Frames composing the flow could not have the same destination as MF and have to leave the flow on v_i. This can be the case, e.g., for unicast traffic only interrupting the flow for one or several vertices on the path. In this paper, we make the assumption that frames leaving the vertex does not affect the compactness of the main flow. This constraint ought be dealt in future works. To take into account the leaving frames, we have to subtract it from Fo obtained using Equation 3. This "new" Fo (noted Fo') is given by:

$$Fo' := [\alpha_o - f_{\alpha_{out}} ; \beta_o - f_{\beta_{out}} ; \gamma_o - f_{\gamma_{out}}] \qquad (4)$$

where α_o, β_o, γ_o are the values of Fo formed using Equation 3, and $f_{\alpha_{out}}$, $f_{\beta_{out}}$, $f_{\gamma_{out}}$ frames leaving v_i and not going to v_{i+1}. For the rest of the paper, we use the notation Fo instead of Fo' to designate the outgoing flow after removing leaving frames.

Once Fo is formed, it becomes Fm for v_{i+1}. For considering the worst-case, we assume that MF is always the last frame of Fm at each vertex. By the way, as soon as MF is queued in a v_i, the outgoing Fo becomes a compact flow due to the serialization effect of the output port (i.e., simultaneously arrived frames are transmitted at the output port one by one, only separated by the Ethernet Inter-Frame Gap). We note Γ_i the set containing all Fc for v_i. A summary illustration of the above flow definitions is given in Figure 4.

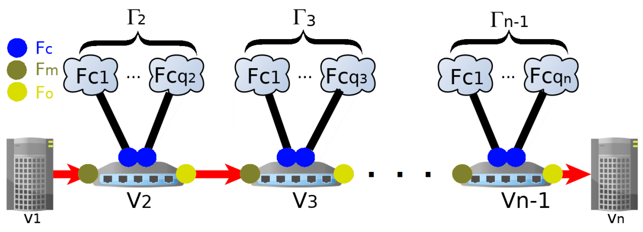

Fig. 4. n vertices with their main flow, concurrent flows and outgoing flow

As a conclusion, a path is composed of n vertices where one Fm and $q_i \times Fc$ compete to access the medium (i.e., same output port link), the result giving Fo. Fm on v_i is determined by Fo on v_{i-1} or by the number of frames coming out the source node if $v_i = v_1$. A concurrent flow is determined by aggregating all possible data frame which may constitute it, as illustrated Figure 5. The precise composition of one Fc depends on the scenario. On each vertex, Fm is disturbed by $q_i \times Fc$ resulting in extra T_{HP} or/and T_{SP} delays for MF.

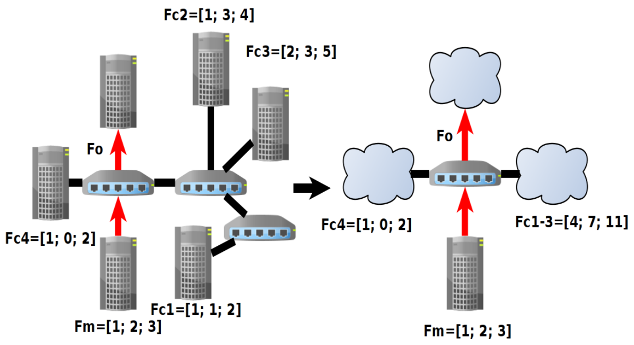

Fig. 5. Flow formation and aggregation

B. *Assumptions and preliminary remarks*

- Our approach is a per-flow and per-path analysis method (see Section III-A), meaning it can be applied independently of the network topology. Of course, our approach can be used for all paths in the network, if the WCD for the whole architecture is needed;
- There is no constraint on the frame transmission offset setting. This means that we can freely determine the offset at source node of each frame belonging to a concurrent flow in order to maximize delays;
- All frames from all end-devices disturb the main flow only once (i.e., no duplication until MF reaches its destination);
- All frames have the same size;
- Frames leaving Fm does not impact the compactness of Fm;
- All interconnection nodes (Ethernet switches) use a Store & Forward policy, with strict priority and a FIFO scheduling within each priority on the output port;
- All link capacities are set to constant data rate C.

IV. Methodology

We present a step-by-step approach to determine T_{HP} and T_{SP}. This term is determined for a frame with a priority that is not the highest. For the sake of readability, proofs of equations provided in this section are put in Appendix.

The methodology is to compute T_{HP} and T_{SP} on each vertex, using its flow characteristics (i.e., α, β and γ values). The computation is based upon the *local analysis*, described in Section IV-B. Local analysis is repeated for each vertex of the path. Once all T_{HP} and T_{SP} are determined, we sum them up to obtain the $T_{HP} + T_{SP}$ term from Equation 1 and finally determine the WCD.

Section IV-C, gives details about the condition that enables us to iteratively determine T_{HP} and T_{SP} terms. Section IV-D proposes an algorithm to automatically compute the T_{HP} and T_{SP} term.

Before starting, we detail the formation of the first Fm, stemming from the source node v_1.

A. Formation of the first Fm

As Figures 4 and 5 suggest, Fm comes to v_2 (the first interconnection node in Figure 5) from the source node v_1. Because IEDs can transmit multiple frames due to multiple running IEC 61850 applications, we can have multiple outgoing frames from the same source. The formation of Fm in v_1 is as follows: all frames are considered as originating from IEC 61850 applications, thus inducing a contention for the access to the network interface card. This contention is equivalent to have MF regarded as Fm and other frames regarded as Fc. T_{HP} + T_{SP} in this situation is the time to transmit all frames disturbing MF. The resulting Fo becomes Fm for the following vertex, i.e., v_2.

B. Local analysis

The Local analysis is composed of two steps:

1 Determine the Theoretical Local Worst-Case Delay (TLWCD, Definition 1);

2 Check if the $TLWCD$ is possible.

1) Determine the TLWCD:

***Definition** 1: The theoretical local worst-case delay (TLWCD) for the main flow Fm on a vertex v_i is the T_{SP_i} + T_{HP_i} delay caused by all frames of all Fc. The $TLWCD$ is defined by:*

$$TLWCD = T_{tr}\sum_{j=1}^{q_i}\gamma_j \tag{5}$$

The $TLWCD$ cannot always be reached, depending on flows α, β and γ values on v_i The next step is to determine if this delay is reachable or not.

2) Check if the $TLWCD$ is reachable: The $TLWCD$ is reachable if and only if the following condition is met:

$$\gamma_{q_i+1} \geq \max_{j\in[1;q_i]}(\beta_j) \tag{6}$$

The proof for Inequation 6 is given in Appendix, Section A. For the rest of the paper, we call the condition given by Inequation 6 the ***TLWCD* condition**.

If the $TLWCD$ condition is met, the worst-case delay for the vertex v_i is found and equal to the $TLWCD$.

If the $TLWCD$ condition is not met, we have to determine a new worst-case delay necessarily lower than the TLWCD. This new worst-case delay is called the Possible Local Worst-case Delay ($PLWCD$, Definition 2).

***Definition** 2: The Possible Local Worst-Case Delay ($PLWCD$) for the main flow Fm on a vertex v_i is the T_{SP_i} + T_{HP_i} delay caused by all Fc giving their α, β or γ values. The $PLWCD$ is defined by:*

$$PLWCD = TLWCD - T_{tr}((\max_{j\in[1;q_i]}(\beta_j) - \gamma_{q_i+1}) \tag{7}$$

The proof of Equation 7 is given in Appendix, Section B. Once the $PLWCD$ is determined, we add it to the WCD obtained from v_{i-1}, giving the WCD for v_i. Fm and Fc are aggregated to form Fo, the latter becoming Fm for v_{i+1}, becoming a compact flow (see section III-A) at the same time. The operation is repeated on the following vertices.

In the following Section, we explain how we can guarantee that the local analysis can be performed on each vertex.

C. Extension to the global path

The previous section highlights the methodology to obtain T_{HP} + T_{SP}. This is done by adding up all T_{HP_i} + T_{SP_i}, which are equal to $TLWCD$ or $PLWCD$. $TLWCD$ and $PLWCD$ are independently computed for each vertex v_i, using the per-node local analysis. This implies that the methodology can be used on v_i regardless of the consequence to obtain the worst-case delay on previous vertices. However, in order to compute the $TLWCD$ and the $PLWCD$ at v_i, frames from different Fc have to reach v_i at a specific date in order for Equation 7 and Equation 5 to be used. Constraints on frame transmission offsets thus appear for each frames on the network.

The methodology proposed in the previous section does not need to know the frame transmission offset settings to be applied. However, we have to prove that such settings exist in order to be able to apply the local analysis on each vertex of the path. This is provided by Theorem 1.

***Theorem** 1: The T_{HP} + T_{SP} term can always be determined by adding all T_{HP_i} + T_{SP_i} determined by the local analysis if there is no constraint about the frame transmission offset at source node for all frames on the network.*

Proof: We first demonstrate the proof on one vertex, then extend it to the overall path.

1) One vertex: We use Figure 6 as an illustration for the one vertex proof, with details explained thereafter.

Let $v_i \in P$ be composed of one Fm and one Fc and tc_1 the date when the first frame f_1 from Fc reaches v_i. Let assume that we are not constrained on the frame transmission offset. Let assume that a given scenario to get the worst-case delay requires that f_1 reaches v_i at a date denoted by ts. We first have:

$$tc_1 = ts$$

If we assume that f_1 needs a duration T_1 to reach v_i from its source node v_1, we can deduce the transmission offset for f_1:

$$offset_1 = tc_1 - T_1$$

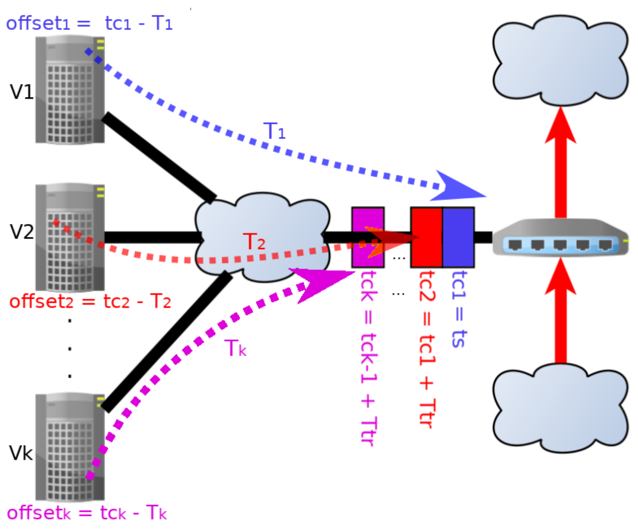

Fig. 6. Frame transmission offset for each frames to obtain a compact flow

In order to apply Equation 5 or 7, respectively to compute $TLWCD$ or $PLWCD$, a compact flow is needed, as demonstrated in their respective proofs. If we define tc_2, $tc_3 \dots tc_k$ the date when frames $f_2, f_3 \dots f_k$ reach v_i and using the definition of compact flow given by Equation 2, we have:

$$tc_k = \begin{cases} tc_1, & \text{if } k = 1 \\ tc_{k-1} + T_{tr}, & \text{otherwise} \end{cases}$$

We can deduce each frame transmission offset for each frame from Fc by:

$$offset_k = tc_k - T_k$$

where T_k is the time for the frame f_k to reach v_i from its source node.

At the end, we only have to know the date ts, which is equal to tc_1. All tc_k values are deduced from tc_1 using a recurrence relation. ts depends on the requirement to achieve the worst-case delay on v_i.

The previous reasoning is independent of other Fc. This is due to the fact that all other frames from other Fc do not reach v_i by the same output port than frames from Fc as there exist only one path between two vertices (definition of G section III-A).

2) Overall path: If there is no constraint on the frame transmission offset, we need two information on each vertex of the path:

- Fm characteristics, i.e., its α, β, γ values;
- the date when the first frame from Fm reaches the vertex.

From the first information, we determine whether $TLWCD$ or $PLWCD$ is used to compute T_{HP_i} and T_{SP_i} on v_i. ts for v_i is determined depending the case. Once ts is determined for v_i, it can be used as reference for all incoming flows competing on v_i.

We can then schedule tc_1 for each Fc to obtain the selected delay with the help of the second information. ■

D. Algorithm

The methodology described in Section IV-B is formalized with Algorithm 1.

Algorithm 1 Tight worst-case delay computation

```
Input: $n$, $Fm$, $\Gamma_1$, $\Gamma_2$, $\dots$ $\Gamma_n$
Output: $WCD$
    Initialisation:
    $WCD = 0$
 1: for $i = 1$ to $n$ do
 2:     if (isTLWCDReachable($\Gamma_i$, $Fm$) == true) then
 3:         WCD = WCD + getTLWCD($\Gamma_i$, $Fm$)
 4:     else
 5:         WCD = WCD + getPLWCD($\Gamma_i$, $Fm$)
 6:     end if
 7:     $sumVertex$ = getSumFlowForVertex($\Gamma_i$, $Fm$, $i$)
 8:     $Fo$ = substractFramesLeavingVertex($sumVertex$, $i$)
 9:     $Fm$ = $Fo$
10: end for
11: return $WCD$
```

Additional remarks:

- The variable WCD represents the $T_{HP} + T_{SP}$.
- We sum all $T_{HP} + T_{SP}$ at each step of the algorithm (line 3 or 5).
- Functions *getSumFlowForVertex()*, *substractFramesLeavingVertex()*, *isTLWCDReachable()*, *getTLWCD()*, *getPLWCD()*, use Equations 3, 4, 5, 6, and 7, respectively.
- The function *substractFramesLeavingVertex()* subtracts all frames from Fo not having the same destination as MF.

V. Application Case

This section propose a concrete example (architecture given in Figure 7) to illustrate our methodology. To set these ideas on the example, we can imagine an architecture similar to the one shown in Figure 1. For this application case, we consider frames with a constant size S and a constant link capacity C for the whole network with a transmission time of $T_{tr} = \frac{S}{C} = 1$. The final result gives $T_{SP} + T_{HP}$ delay which must be added to T_{LP} and T_{tr}. To be in the worst-case, we assume that low priority blocking resulting in T_{LP} systematically occurs at each vertex v_i excepted the last one (destination node). T_{tr} occurs on each edge (i.e., network links) traversed by the frame. As an IEC 61850 architecture, we consider MF being transmitted by a low-priority protocol, e.g., MMS. Therefore, HP frames belongs either to SMV or GOOSE

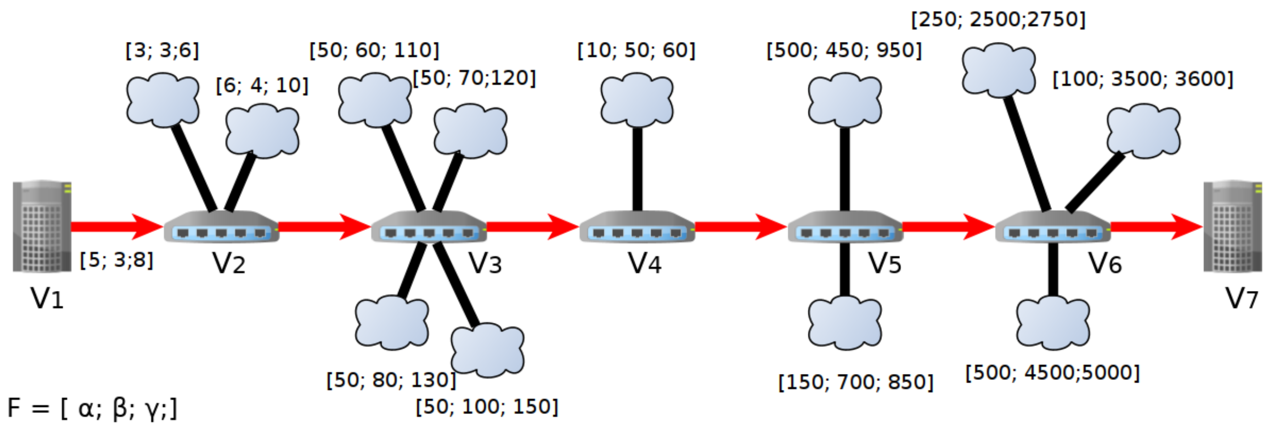

Fig. 7. Architecture under study

	Vertices (v_i)					
	v_1	v_2	v_3	v_4	v_5	v_6
Incoming flows						
$\boldsymbol{Fm}$	[0; 1; 1] (MF)	[5; 3; 8]	[14; 10; 24]	[214; 320; 534]	[224; 370; 594]	[874; 1520; 2394]
$\boldsymbol{q_i}$	0	2	4	1	2	3
Local analysis results						
$\boldsymbol{TLWCD_i}$ (Equation 5)	7	16	510	60	1800	11350
$\boldsymbol{\gamma_{q_i+1}}$	-	8	24	534	594	2394
$\boldsymbol{max(\beta_j)}$ ($j \in [1;q_i]$)	-	4	100	50	700	4500
$\boldsymbol{\gamma_{q_i+1} \geq \max_{\beta_j}}$ ($j \in [1;q_i]$) (Inequation 6)	-	**true**	false	**true**	false	false
$\boldsymbol{PLWCD_i}$ (Equation 7)	-	-	434	-	1694	9244
Outgoing flow and WCD						
$\boldsymbol{WCD_i}$	7	23	457	517	2211	**11455**
$\boldsymbol{Fo}$ ($Fm + \sum_{j=1}^{q_i} F_{cj}$, Equation 3)	[5; 3; 8]	[14; 10; 24]	[214; 320; 534]	[224; 370; 594]	[874; 1520; 2394]	[1724; 12020; 13744]

TABLE IV
RESULTS WITH ALGORITHM 1 WITH THE EXAMPLE ARCHITECTURE GIVEN FIGURE 7

and LP frames belongs to best-efforts protocols. In order to simplify the analysis, we consider that all frames of all Fc have the same destination as MF. We consider that the periodicity of all flow is equal to the WCD.

Results are summarized in Table IV.

v_1 is composed of MF and 7 concurrent frames generated by IEC 61850 applications. In the worst-case, MF has to wait the transmission of all concurrent frames before being itself transmitted. As MF has been disturbed by 7 frames, the WCD at v_1 is set to 7. Fo is subsequently formed by the aggregation of all frames from v_1.

On v_2, Fm can be disturbed by all Fc frames, according to $TLWCD$ condition. The WCD is incremented by $TLWCD$, giving a WCD of 23 at v_2.

v_3 becomes the first vertex where $TLWCD$ condition is not met making it impossible to obtain the TLWCD. We then compute the $PLWCD$ using Equation 7 and add it to the current WCD. The result becomes 457.

v_4 is in the same situation as v_1, i.e., the condition to obtain $TLWCD$ is met. The WCD is incremented by the $TLWCD$ obtained on v_4, giving 517 as the result.

v_5 and v_6 lead to the same situation, namely $TLWCD$ is not possible, thus computing the $PLWCD$ and add it to the WCD. The final result gives 11455, which represents $T_{HP} + T_{SP}$ for the given path. Using Equation 1, we can compute the WCD by adding up T_{LP} and T_{tr}:

$$WCD = T_{HP} + T_{SP} + \sum_{i=1}^{n-1} T_{LP} + (n-1)T_{tr}$$
$$WCD = 11455 + T_{tr}(n-1) + (n-1)T_{tr}$$
$$WCD = 11455 + 1(6-1) + 1(6-1)$$
$$WCD = 11465$$

WCD gives a tight representation of what the worst-case delay for an MMS frame is in case of the architecture given figure 7. As long as the periodicity of all Fc in the network is greater than or equal the WCD, the latter can be guaranteed. This information could be used by the IEC 61850 engineer, e.g., to configure the IEC 61850 architecture accordingly.

Note that operations used to obtain the WCD are boiled down to simple operations at each node (i.e., comparison, sum and maximum operations), reducing thus the complexity and allowing the scalability of our analysis method.

VI. Conclusion, future works

In this paper, we proposed a new approach to determine a tight WCD for switched Ethernet running strict-priority policy, with a focus on supporting IEC 61850 protocols requiring different real-time constraints. This approach is based on local worst-case delay analysis and an iterative algorithm, working with flows modeled as a 3-tuple object representing the quantity of same-priority, high priority and total number of frames.

Through this modelling, we deduced several properties used to determine a tight end-to-end WCD, depending on the number of frames of different priorities which compose a flow. All the analysis remains upon the possibility to schedule all transmission offsets at source nodes for each frame of the network, allowing us to gives the worst-case scenarii in which the WCD is encountered. As we do not focus on the determination of these offsets, we simply consider them as possible, only searching for the WCD. Finally, the determination of the WCD is scalable, as we only need to perform basic operations such as comparison or sum on each path vertex. The complexity depends on the number of vertices and the number of Fc for each vertex. An algorithm is also provided implementing the methodology. Although this work is motivated by the IEC 61850 application context, the proposed approach can be used in other use cases using Ethernet with strict-priority supported by IEEE802.1Q.

Our future work will focus on extending our method to take into account the multiple frame sizes. We also aim to integrate this work into our previous analysis made in [15] in order to obtain a complete analysis of the WCD in a IEC 61850 context with an integrated TSN technology.

Appendix

A. Proof of Inequation 6

Determination of T_{SP_i}: Let us assume that all incoming flows on v_i are compact and that the first frame from each flow reaches v_i at the same time. Knowing that Fm competes with q_i Fc, we can deduce that each SP frame from Fm is disturbed by q_i frames for each T_{tr}. Moreover, if we make the assumption that SP frames are first transmitted by all Fc and $\gamma_{q_i+1} \geq \max_{j\in[1;q_i]}(\beta_j)$, then all frames of Fm are disturbed by all SP frames from the largest Fc. All SP frames from all Fc disturb Fm. Consequently, T_{SP_i} is given by:

$$T_{SP_i} = T_{tr}\sum_{j=1}^{q_i}\beta_j$$

If $\gamma_{q_i+1} < \max_{j\in[1;q_i]}\beta_j$, some frames coming from Fc with the highest β value will not disturb MF from Fm as the latter will be either already transmitted or stored in v_i.

Determination of T_{HP_i}: Assume that the first HP frames from all Fc start to reach v_i once the last SP frame from all Fc are stored. If all HP frames from all Fc form a compact flow, all HP frames block SP frames from Fm that are not yet transmitted. T_{HP_i} is given by:

$$T_{HP_i} = T_{tr}\sum_{j=1}^{q_i}\alpha_j$$

Finally, MF has to wait $T_{SP_i} + T_{HP_i}$:

$$T_{tr}(\sum_{j=1}^{q_i}\alpha_j + \sum_{j=1}^{q_i}\beta_j) = T_{tr}\sum_{j=1}^{q_i}(\alpha_j+\beta_j) = T_{tr}\sum_{j=1}^{q_i}\gamma_j$$

which corresponds to the definition of $TLWCD$ given by Equation 5. This waiting time is only possible if $\gamma_{q_i+1} \geq \max_{j\in[1;q_i]}(\beta_j)$, leading to $TLWCD$ condition.

B. Proof of Equation 7

We determine both T_{HP_i} and T_{SP_i} to obtain WCD.

Determination of T_{SP_i}: We search to maximize the blocking time caused by SP frames from Fc to Fm on v_i. The blocking time caused by SP frames to other SP frames can be of two kinds:

- direct blocking: SP frames from one or multiple Fc reach v_i at the same time than a SP frame from Fm and is the last served;
- indirect blocking: SP frames from one or multiple Fc are already queued on v_i, blocking in turn frames from Fm because of a FIFO scheduling policy for SP frames.

HP frames from Fm cannot be blocked by SP frame. However, HP frames always block all SP frames from other Fc, which leads to an indirect blocking delay for the following SP frames from Fm. Thus, the blocking time is the same regardless the priority of the the frame from Fm.

If $\gamma_{q_i+1} < \max_{j\in[1;q_i]}(\beta_j)$ (Inequation 6 not met), it is impossible for Fm frames to be disturbed by all frames from all Fc.

To maximize T_{SP} we can consider the following situation. Firstly, we assume that the last frame from Fm reaches v_i at the same date, denoted by t_{last}, than the last frame from the concurrent flow having the highest β value. This leads to the situation that all frames from Fm are disturbed by SP frames from the Fc with the highest β value, causing direct blocking. We note by t_{first} the date when the first frame from Fm reaches v_i. We deduce the following relation:

$$t_{last} = t_{first} + \gamma_{q_i+1} \times T_{tr}$$

Secondly, we assume that all Fc start to transmit their SP frames at a same date $t < t_{first}$. This leads to accumulate SP frames from all Fc while frames from Fm have not yet reached v_i, causing indirect blocking. We note by:

$$\Delta = t_{first} - t$$

the duration between the arrival of all first frames from Fc and the arrival of the first frame from Fm. All frames transmitted during Δ therefore do not disturb Fm.

Knowing that the number of frames transmitted between $t_{last} - t$ is given by:

$$\max_{j\in[1;q_i]}(\beta_j) = \frac{t_{last} - t}{T_{tr}}$$

and that the number of frames transmitted between $t_{last} - t_{first}$ is given by:

$$\gamma_{q_i+1} = \frac{t_{last} - t_{first}}{T_{tr}}$$

we can deduce that:

$$\Delta = t_{first} - t$$
$$\Delta = (-T_{tr} \times \gamma_{q_i+1} + t_{last}) - (-T_{tr} \times \max_{j\in[1;q_i]}(\beta_j) + t_{last})$$
$$\Delta = T_{tr}(\max_{j\in[1;q_i]}(\beta_j) - \gamma_{q_i+1})$$

Apart from frames transmitted during Δ, all SP frames from all Fc disturb frames from Fm. T_{SP_i} is thus given by:

$$T_{SP_i} = T_{tr}(\sum_{j=1}^{q_i} \beta_j) - \Delta$$
$$T_{SP_i} = T_{tr}(\sum_{j=1}^{q_i} \beta_j - (\max_{j\in[1;q_i]}(\beta_j) - \gamma_{q_i+1}))$$

where $\sum_{j=1}^{q_i} \beta_j$ is the sum of all SP frames from all Fc.

Determination of T_{HP_i}: Let us assume that the first HP frames from all Fc start to reach v_i once the last SP frame from all Fc are stored. If all HP frames from all Fc form a compact flow, all HP frames block SP frames from Fm that are not yet transmitted. T_{HP_i} is given by:

$$T_{HP_i} = T_{tr} \sum_{j=1}^{q_i} \alpha_j$$

From our previous analysis, we can sum T_{HP_i} and T_{SP_i}:

$$PLWCD = T_{tr}\Big[\sum_{j=1}^{q_i} \alpha_j + \sum_{j=1}^{q_i} \beta_j - (\max_{j\in[1;q_i]}(\beta_j) - \gamma_{q_i+1})\Big]$$

Observing that:

$$TLWCD = T_{tr} \sum_{j=1}^{q_i} \gamma_j$$
$$= T_{tr} \sum_{j=1}^{q_i} (\alpha_j + \beta_j)$$

we have:

$$PLWCD = TLWCD - T_{tr}((\max_{j\in[1;q_i]}(\beta_j) - \gamma_{q_i+1})$$

which corresponds to the definition of $PLWCD$ given by Equation 7. Note that if:

$$\gamma_{q_i+1} = \max_{j\in[1;q_i]}(\beta_j)$$

we have:

$$PLWCD = TLWCD$$

which is consistent with Inequation 6.

References

[1] J.D. MacDonald. *Electric Power Substations Engineering*. CRC Press, 2012.

[2] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid - the new and improved power grid: A survey. *IEEE communications surveys & tutorials*, 14(4):944–980, 2012.

[3] International Electrotechnical Commission. Communication networks and systems for power utility automation — part 5: Communication requirements for functions and device models. *IEC Standard*, 61850, 2013.

[4] K. Tindell and J. Clark. Holistic schedulability analysis for distributed hard real-time systems. *Microprocessing and microprogramming*, 40(2-3):117–134, 1994.

[5] P. Pop, P. Eles, and Z. Peng. Schedulability analysis and optimisation for the synthesis of multi-cluster distributed embedded systems. *IEE Proceedings-Computers and Digital Techniques*, 150(5):303–312, 2003.

[6] H. Charara, J-L. Scharbarg, J. Ermont, and C. Fraboul. Methods for bounding end-to-end delays on an afdx network. In *IEEE ECRTS'06*, pages 10–pp, 2006.

[7] M. Adnan, J-L. Scharbarg, J. Ermont, and C. Fraboul. An improved timed automata approach for computing exact worst-case delays of afdx sporadic flows. In *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, pages 1–8. IEEE, 2012.

[8] J-P. Georges, T. Divoux, and E. Rondeau. Strict priority versus weighted fair queueing in switched ethernet networks for time critical applications. In *19th IEEE International Parallel and Distributed Processing Symposium*, pages 8–pp. IEEE, 2005.

[9] Hanno Georg, Nils Dorsch, Markus Putzke, and Christian Wietfeld. Performance evaluation of time-critical communication networks for smart grids based on iec 61850. In *INFOCOM, 2013 Proceedings IEEE*, pages 3417–3422. IEEE, 2013.

[10] H. Bauer, J-L. Scharbarg, and C. Fraboul. Applying trajectory approach with static priority queuing for improving the use of available afdx resources. *Real-time systems*, 48(1):101–133, 2012.

[11] X. Li, J-L. Scharbarg, and C. Fraboul. Analysis of the pessimism of the trajectory approach for upper bounding end-to-end delay of sporadic flows sharing a switched ethernet network. In *RTNS*, pages 149–158, 2011.

[12] J. Diemer, D. Thiele, and R. Ernst. Formal worst-case timing analysis of ethernet topologies with strict-priority and avb switching. In *7th IEEE International Symposium on Industrial Embedded Systems (SIES'12)*, pages 1–10. IEEE, 2012.

[13] T. Docquier, Y. Q. Song, V. Chevrier, L. Pontnau, and A. Ahmed-Nacer. Generating substation network simulations from substation configuration description files. In *13th Junior Researcher Workshop on Real-Time Computing, RTNS*, 2019.

[14] H. León, C. Montez, M. Stemmer, and F. Vasques. Simulation models for iec 61850 communication in electrical substations using goose and smv time-critical messages. In *IEEE WFCS*, pages 1–8, 2016.

[15] T. Docquier, Y. Q. Song, V. Chevrier, L. Pontnau, and A. Ahmed-Nacer. Iec 61850 over tsn: traffic mapping and delay analysis of goose traffic. In *IEEE ETFA*, sep. 2020 (to appear).

[16] J. Rox and R. Ernst. Formal timing analysis of full duplex switched based ethernet network architectures. Technical report, SAE Technical Paper, 2010.