# A methodology for the evaluation of the message transmission delay over IEC 61850 communication network - a real-time HV/MV substation case study

Ronak Feizimirkhani, Antoneta Iuliana Bratcu, Yvon Bésanger, Antoine Labonne, Thierry Braconnier

# A methodology for the evaluation of the message transmission delay over IEC 61850 communication network – a real-time HV/MV substation case study

Ronak Feizimirkhani[1,2], Antoneta Iuliana Bratcu[1], Yvon Bésanger[2],
Antoine Labonne[2], Thierry Braconnier[2]

[1]Univ. Grenoble Alpes, CNRS, Grenoble INP (Institute of Engineering Univ. Grenoble Alpes),
GIPSA-lab, 38000 Grenoble, France
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP (Institute of Engineering Univ. Grenoble Alpes),
G2Elab, 38000 Grenoble, France

*Abstract*— **The growing intelligence of the nowadays distribution grids is due to the integration of Information and Communication Technologies (ICTs). Such a development is converting the traditional grid into the smart grid concept. Communicating over a reliable ICT infrastructure, distributed Intelligent Electronic Devices (IEDs) perform supervision, protection, and control of the smart grid. The communication network is not errorless, and any error of the underlying communication network can degrade the power grid performance or even cause instability in it. Hence, there is a necessity to obtain a modeling method to analyze its Quality of Service (QoS). One of the most important quality measures of the Substation Communication Network (SCN) traffic is the message transmission delay. Thus, this paper proposes a methodology to evaluate the message transmission delay through an analytical model. To estimate the transmission delay, a port connection model represents the message distribution model. Then a traffic-flow-source model, and a traffic-flow-service model are applied to characterize each flow while it is served through different ports. Finally, delay is calculated based on the corresponding flow and service information. The estimation model is applied on an IEC-61850-MMS (Manufacturing Message Specification)-based communication scenario for a load-shedding goal that is considered over a real reduced-scale HV/MV smart substation test bench. The estimated results are compared to the measured values by the Wireshark network analyzer that indicates an estimation error of less than 10%.**

*Keywords*— **Smart grid, IEC 61850 MMS, Network Calculus Theorem, maximum message transmission delay**

## 1 Introduction

Modern power grids, *i.e.*, smart grids, are more and more using ICT features to do monitoring, control, and protection for better production and efficiency. In addition, spatially Distributed Energy Resources (DERs) are increasingly used to enhance reliable and clean energy production. To this end, Intelligent Electronic Devices (IEDs) are vastly integrated to cooperate efficiently, and ICT is widely used as an infrastructure for all different entities to communicate [1]. The interactions between the power grid and the communication network should be rethought to be able to provide a flexible, available, and secure production-consumption system for the future smart grid. In this order, as ICT is not error-free, it is required to consider the characteristics of the communication network for the analysis of the power system as a whole. HV/MV substations are the main elements of modern power systems, as interfaces between transmission and distribution grids. As a part of the smart grid, a need for more intelligence in these substations leads to the development of Substation Automation Systems (SAS) supported by ICT-based Substation Communication Networks (SCN) [2], [3]. Thus, SAS is growingly dependent on SCN for monitoring, control, and protection. In this case, building a custom-designed SCN can help to maintain a fast and reliable information transmission. Hence, reliable SCN models are of the essence to allow network performance assessment under different conditions.

Transmission delay, as one of the most important criteria of network performance evaluation, can result in an abnormal operation of the power grid and thus, has recently attracted the attention of many researchers. Different methods have been developed to evaluate the IEC 61850 SCN traffic behavior [4], [5], [6], [7], [8]. [9] introduces OPNET modeler as a test bench for analyzing the communication network performance where three types of IEDs are modeled: breaker IED, Merging Unit (MU), and a Protection and Control IED (P&C). The result shows that the network performance satisfies the time-critical messages in SAS. In [10], the authors performed an average and worst-case message delay analysis of intra- and extra-bay level communication for different sampling rates. In this work, the communication network including a substation and DERs is simulated, in which wired and

wireless communication solutions are compared. [11] presents a simulation of the Sampled Value (SV) traffic generation in NS3 open-source network simulator to provide also a test bench for performance analysis. The related results are validated using the Wireshark network analyzer. The proposed model can be used to evaluate different aspects from critical to safety utility applications.

The authors of [12] present a modeling approach for evaluating the real-time capability of communication technologies for smart grid applications. A simulation model along with an analytical approach is developed based on the Network Calculus Theorem. The overall real-time performance study of the bay Local Area Network (LAN) is done to examine the generation of SV and Generic Object-Oriented Substation Event (GOOSE). The simulation is done by the OPNET modeler. Delay values satisfy the real-time requirements defined in IEC 6185.

[13] investigates the communication delay in System Integrity Protection Schemes (SIPS), which extends the protection from the local equipment to the integrity of the power system. Instead of constant or stochastic models often found in the literature, a dynamic bounded-delay estimation model is proposed based on the Network Calculus Theorem. The bounded model suggests the worst-case performance, first, over wide-area protection, second, over substation-area protection system. Moreover, the proposed model is verified by application over IEC 61850 T2-2 substation system, IEEE 14 bus system, and China Southern Power Grid SDH system.

A combination of Network Calculus and measurements is proposed in [14] to construct a real service curve model, and produce accurate delay bounds that can be validated against the measured values over a real Ethernet network of IEC 61850. In this approach, the latency component of service curve is extracted based on measuring values, namely by injecting sampled value messages at low rate and measuring the maximum delay value which is the non-queuing delay (*i.e.*, physical delay property of the related switch).

Delays in IEC 61850 process bus are characterized using an analytical delay estimation methodology and described statistically through simulations in [15]. This paper considers both a steady-state traffic behavior and the traffic resulting from a breaker failure event. Result analysis shows that delay estimation is a time-consuming task and the analytical method is a better option to find the worst-case delay. In addition, the scalability analysis provides a more significant delay as the size of the substation increases.

[16] proposes an IEC 61850 over the Time-Sensitive Networking (TSN) mapping, based on the characteristics of both IEC 61850 protocols and TSN traffic classes. The main goal is to map SV protocol on a TSN-protected class to provide the minimum guaranteed delay performance and jitter property. It also provides an exact worst-case delay bound of GOOSE protocol belonging to the trip and blocking messages transfer. The simulation is performed using the OMNeT++ simulator along with IEC 61850 and TSN models.

Among these papers, none of the above deals with a comprehensive methodology that estimates the transmission delay through a self-contained and easy-to-implement procedure. Therefore, this paper develops a comprehensive delay evaluation methodology to meet this need. The considered methodology requires a data-transmission model over SCN which is developed using an analytical model proposed in [17] based on the Network Calculus Theorem. Unlike in the paper taken as our reference -- supposing the characteristics of all the data flows are known -- in this work, the understudying traffic flows are totally unknown. Hence, we propose an identification step to find the corresponding arrival curve coefficients. In addition, the test bench is here plugged on a real reduced-scale distribution grid. This work also aims at obtaining an analytical method to be further integrated into a network-traffic control design, delay mitigation methods, or network configuration optimization. Here, the proposed algorithm is applied on a type of data (MMS) mostly representing the overall characteristics of the IEC 61850 protocols.

The remainder of this paper is divided into four sections. Section 2 explains the applied analytical model, and the Network Calculus Theorem principles, that are used to model the message distribution, and further to calculate the maximum delay. In Section 3, the delay estimation algorithm is explained in detail as it is applied to the considered experimental test case. Section 4 summarizes the results and proposes a general methodology for delay estimation. Also, the estimation results are validated against the measured ones. Section 5 provides the conclusion and related perspectives.

# 2 Analytical modeling: Network Calculus Theorem [17], [18], [19]

To improve the operation and management of a SAS and to have a fast reliable information transmission, it is necessary to build a well monitored and custom-designed SCN. Hence, reliable SCN models are required to be able to evaluate the network performance under different conditions. One of the most critical issues of network reliability is the real-time performance analysis [20]. A delay of message transmission results in an improper information delivery, which may lead to poor operation in the smart grid, even instability depending on the situation.

To configure a proper SCN, a designing plan can be developed based on the maximum delay estimation, the traffic load distribution under different network conditions, and equipment selection. This can help to find the best transmission path through which messages are served with the minimum delay. Generally, four types of delay are identified inside a switch: 1) packet receiving delay, 2) processing delay, 3) queueing delay, 4) transmission delay. The first and second types are considered constant values around 3 $\mu$s per switch [21]. The third and fourth types

are variable and need to be estimated.

Network calculus theorem as a methodology helps to evaluate the communication network performance through quantitative indicators such as network services assigned to the flows, window flow control, scheduling and buffering, or delay dimensioning [22], [23], [24]. Network calculus defines a cumulative function $F(t)$ that describes a data traffic flow by the number of bits observed on the flow in the time interval $[0, t]$. Any arriving flow needs a guaranty to be served at each port, so it is upper-bounded by an arrival curve and lower-bounded by a service curve [25].

**Arrival curve,** as a model, defines constraints on the traffic flow arriving at the service provider, *e.g.,* Ethernet switch. Such a limit is upper-bounded by an increasing function $\alpha(t)$, named arrival curve, if and only if [19]:

$$F(t) - F(s) \leq \propto (t - s), \forall\, s \leq t, \tag{1}$$

where $\alpha(t)$ is called the arrival curve for $F(t)$. One of the most used arrival curves is $(\sigma, \rho)$-model proposed by Cruz [24], which is a simple linear model:

$$\propto (t) = \sigma + \rho t, \tag{2}$$

where $\sigma$ signifies the burstiness of the flow in bits and $\rho$ represents an upper bound on the long-term average rate of the traffic flow in bits/s. The flow $F(t)$ is $(\sigma, \rho)$-upper-bounded if and only if [19]:

$$F(t) - F(s) \leq \rho(t - s) + \sigma, \forall\, s \leq t \tag{3}$$

**Service curve** defines lower bounds on the services provided by the servers as Ethernet switches or routers. The service curve model is a function of time that specifies the services provided by those service units during a defined time interval. The rate-latency $\beta(t)$ model is a widely used service curve model represented by a linear function as below:

$$\beta(t) = R[t - T]^{+}, \tag{4}$$

where $[\mathrm{x}]^{+}$denotes the $\max\{\mathrm{x}, 0\}$, $R$ signifies the transmission rate in bits/s, and $T$ represents the latency in s [19]. When a flow receives $\beta_{R,T}$ as a service curve, it means that it is served by the rate of $R$ in $T$ time after its arrival at the service system. Network calculus is developed in the min-plus algebra, which characterizes the amount of traffic and available service as functions of time [26].

**Min-plus algebra** defines an algebraic structure that changes the most common conventional math operators on the set of integers, $\mathbb{Z}$, or reals, $\mathbb{R}$, as follows: addition to minimum (min) computation, and multiplication to addition (plus). The maximum delay is estimated such that the predefined constraints of Network Calculus are satisfied. Related constraints are based on the min-plus algebra, so it is necessary to know the min-plus convolution and its dual deconvolution functions. If $f$ and $g$ are two functions, then their min-plus convolution is defined as below [27]:

$$(f \otimes g)(t) = inf_{0 \leq s \leq t}\{f(t - s) + g(s)\}, \tag{5}$$

while for $t < 0$, $(f \otimes g)(t) = 0$. For the same functions $f$ and $g$, min-plus deconvolution is defined as below [27]:

$$(f \oslash g)(t) = sup_{s \geq 0}\{f(t + s) - g(s)\} \tag{6}$$

Based on min-plus algebra, by replacing $\alpha$ and $\beta$ in (6), the bounds on the output flow from a service system, $\alpha^{*}$, can be computed by the deconvolution of arrival curve and service curve [19]:

$$\alpha^{*} = (\alpha \oslash \beta)(t) = sup_{\forall s \geq 0}\{\alpha(t + s) - \beta(s)\} \tag{7}$$

Delay $D(t)$ for the flow at the time $t$ is upper-bounded by:

$$D(t) \leq h(\alpha, \beta) = sup_{\forall s \geq 0}\{inf\{\tau \geq 0 : \alpha(s) \leq \beta(s + \tau)\}\}, \tag{8}$$

where $sup\{S\}$ and $inf\{S\}$ are respectively the least upper bound and greatest lower bound of a subset $S$, and $\alpha$ and $\beta$ are the arrival and service curves, respectively. In particular, if $\alpha(t) = \rho t + \sigma$, and $\beta(t) = R[t - T]^{+}$, then $\alpha^{*}$ also has the $(\alpha^{*}, \beta^{*})$ type with $\rho^{*} = \rho$, and $\sigma^{*} = \sigma + \rho T$ [19]. So, the maximum delay is calculated as:

$$D_{ij}(t) \leq h(\alpha_i, \beta_j) = T_i + \sigma_j / R_i, \tag{9}$$

where the maximum message transmission delay is obtained for the flow *j*, passing through each port *i*, and depends on the characteristics of its arrival curve, *i.e.*, $\sigma_j$, and the received service from the port, *i.e.*, $T_i$ and $R_i$.

Next section details the methodology for the evaluation of maximum delay over a real reduced-scale grid. Thus, considering a real-world case, the employed step-by-step procedure allows for both introducing the main modeling notions and formulas, meanwhile immediately illustrating them on the considered scenario.

# 3 Maximum message transmission delay

Four main concepts of the analytical model that shape the SCN transmission delay are: *physical connection model*, *logical connection model*, *source model*, and *service model* in the form of matrices [23], using an algorithm for building the information flow distribution matrix, supposing a completely known communication network and parameters. The message distribution matrix is then used for maximum delay estimation on each traffic flow using (9).

## 3.1 Experimental communication scenario

In this research, the delay estimation process is applied to a communication scenario with the goal of load shedding in a distribution system. The generated traffic by a real reduced-scale SCN is perturbed by some flows transmitted through the network as background traffic. The experiment is performed on a real-time test bench as shown in Figure 1, consisting of a real HV/MV smart substation at a reduced scale equipped with real electrical devices (*e.g.*, IEDs marked in red).

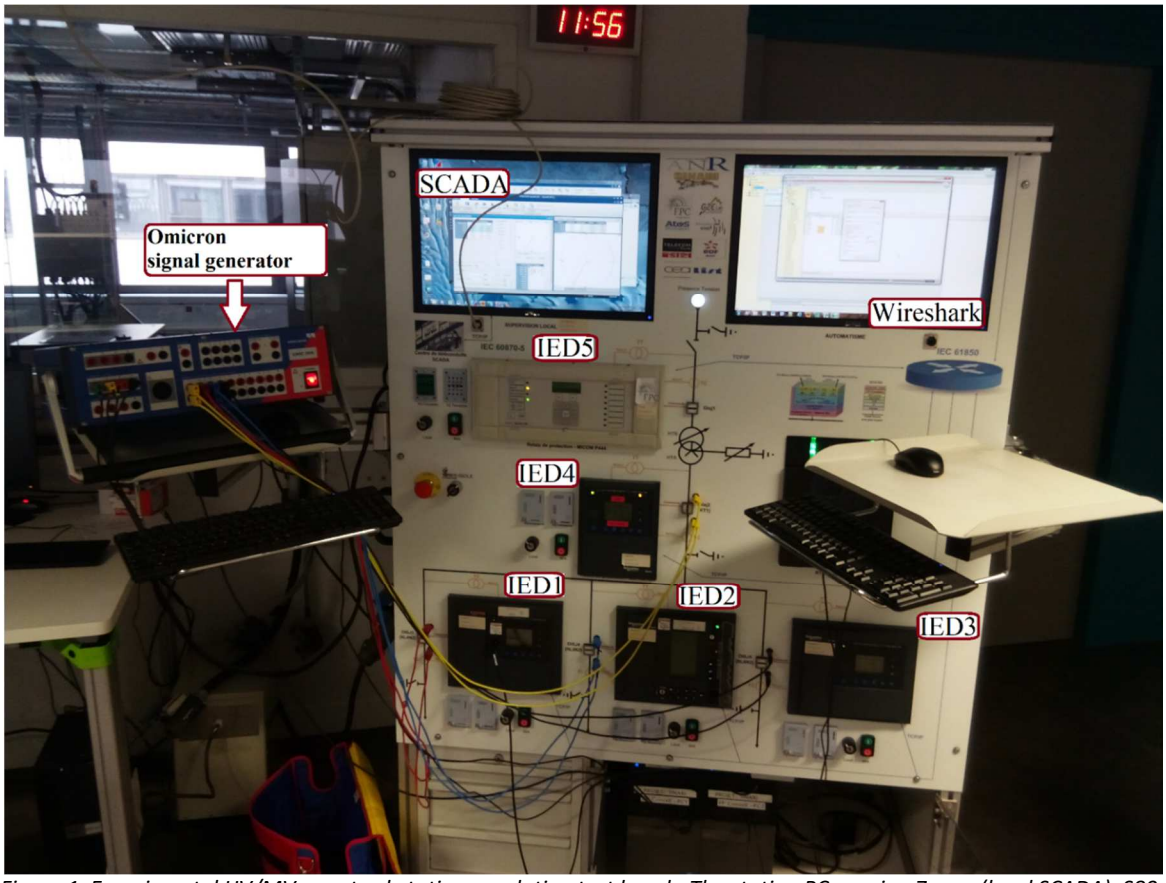



*Figure 1 Experimental HV/MV smart substation-emulation test bench. The station PC running Zenon (local SCADA), S80-Sepam, S40-Sepam, and P444 are all installed on this bench.*

As illustrated in Figure 2, the test bench is connected on a real lab-scale distribution grid of 30 kW under 400 V. It represents a real MV distribution system of 30 MW – 20 kV with conservation of the scale factor. In this scenario, three feeders (L4 to L6) send their typical active power (kW) consumption measured by IEDs to the supervisory unit. IED1 and IED2 are connected to constant-active-power loads, and IED3 is connected to a variable typical load of an urban area. The active power consumption profile measured by IED3 is directly created by a Premium PLC. Load units communicate through their related S84-Sepam Power Line Communication (PLCs) as IEDs. Critical measured values (*i.e.*, line voltages or active power) are sent to a Zenon supervisory and control application unit (Zenon is a local SCADA in this case), which runs on the PC station as the control center. IED1 is connected to a fixed consumer (CH3) on L4, and its active power profile is equal to $P = 2.6$ kW.

The active power consumer (CH2) of IED2, which is connected to L5, is $P = 3$ kW. For IED3, connected to L6, a variable active power curve is defined as the related consumption (CH1), which is shown in Figure 3: a typical 24-hour active power consumption curve with a 10-minute time step (thus 144 points) is simulated with a reduced time-scale curve of 24 minutes with a 10-s time step.

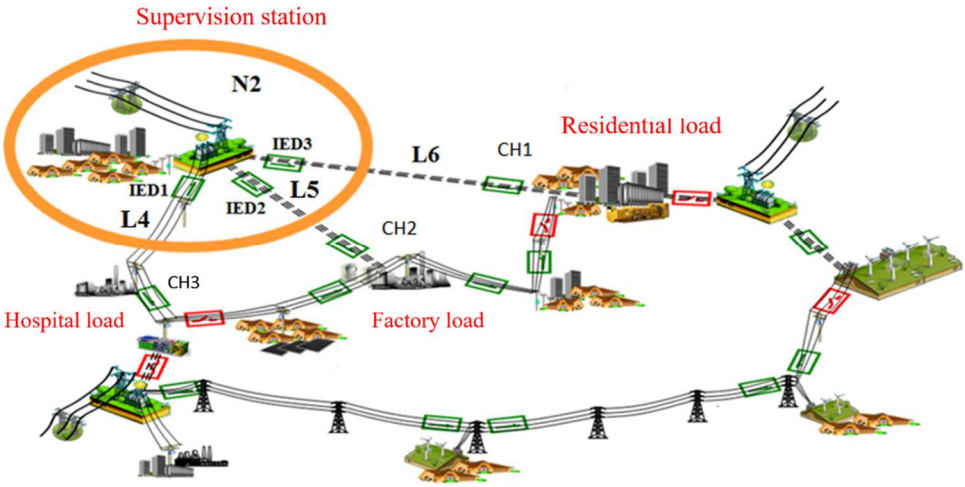



*Figure 2 Reduced-scale lab distribution grid with the smart substation. All the introduced loads, and their connected digital protection relays participating in the load-shedding scenario -- i.e., IED1, IED2 and IED3 -- are illustrated.*

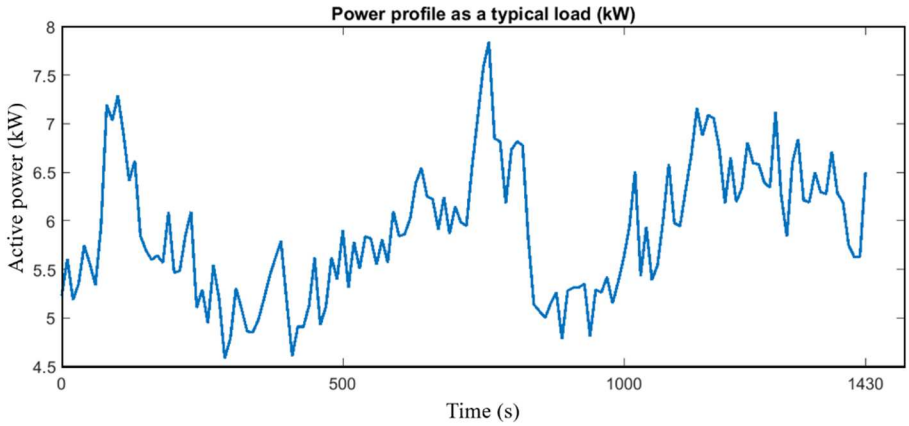



*Figure 3 A typical 24-hour active power consumption curve with a 10-minute time step is predefined to be the residential load consumption as a reduced time-scale curve of 24 minutes with a 10-s time step.*

There is only one VLAN group in the considered scenario that includes two switches: CISCO (2950) and NTRON (516TX). All the messages are of the MMS type that allows MMS client such as SCADA to access all the IED objects. MMS is based on the TCP/IP protocol that ensures the reception of a message by a handshaking protocol from the receiver. Power grid is assumed to be stable and faultless, so the communication flow is assumed to obey some regular behavior pattern, without exhibiting extreme or discontinuous phenomena. Traffic flows normally, and there is no corruption, loss, message duplication or congestion in the understudying communication scenario. Therefore, nor congestion control is applied nor retransmission mechanisms [28].

An active power threshold is predefined at SCADA, and it is compared to the sum of measured power values received from IEDs. If it overpasses the threshold, SCADA decides to shed the load with the lowest priority. Respectively, the connected loads to the IED1, IED2, and IED3 are assumed to be a residential, a hospital, and a factory load, respectively. Thus, IED3 receives the command to open L6, and the load on this line is then no more supplied. So, the other loads, with higher priorities, are sure to receive electricity properly. Threshold setting, load-shedding program, and the required calculations are done in SCADA (*i.e.*, Zenon application).

The related schematic of the considered communication scenario is represented in Figure 4 – without background traffic. IEDs and SCADA are connected using an Ethernet switch, measured values are sent to SCADA, and the related load shedding command is sent from SCADA to IED3. IEC 61850 MMS protocol is used for all messages.

Two IEDs (S40-Sepam and P444) are added to the network generating two perturbation MMS flows (some measured data not involved in the load shedding scenario) in order to integrate a background traffic, i.e., IED4 and IED5. So, six flows are transmitted: five flows from IEDs to SCADA, and one flow from SCADA to IEDs. The network configuration is shown in Figure 5.

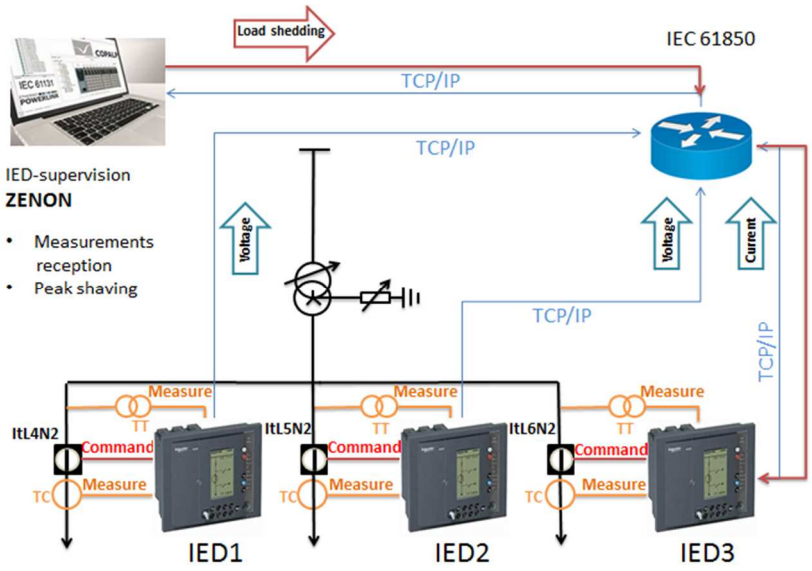



*Figure 4 Load shedding control process in Zenon application (i.e., SCADA) on PC. The communication trajectories are indicated in blue for the measurements to SCADA and red for the load shedding control command to IED3.*

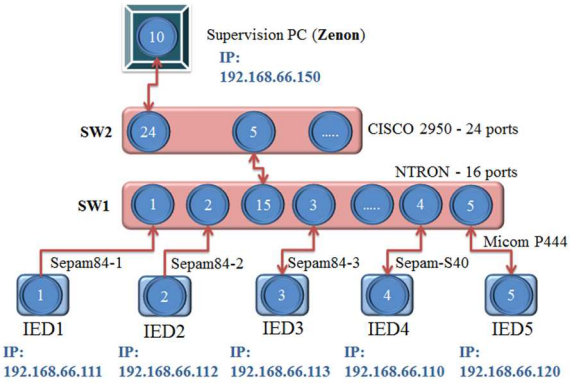



*Figure 5 Communication network configuration perturbed by two traffic flows.*

Primary matrices must be defined to start the message distribution algorithm, which forms the structure of the message distribution model. The first step of maximum delay estimation is to model the logical and physical port connections. Hence, the port connection model, service and source models are presented in the next sections.

## 3.2 Port connection models

Ports of devices in SCN are independent units. The transmission path of each of the information flows from source to destination is a combination of such ports. Each port can send, forward, or receive a message through this path. Here, port connection models, including a physical connection model and a logical connection model, are represented to describe the relationship of the ports on the considered path. Physical transmission media such as cables or optical fibers represent physical connections, and the control logics inside switches are defined as logical connections. When a message arrives at a switch port, control logic of the switch decides which port it should be sent to.

Above all the ports of the considered communication network are labeled by a number from 1 to 14, and they are denoted by nodes. Dashed arcs represent a logical connection (L), and solid arcs represent a physical connection (P). In addition, each arc is directed to show the transmission direction. Building the message distribution model begins by drawing all the connections (cabled or switch logic), as is shown in Figure 6. Distribution algorithm is initialized by definition of the main elements of the corresponding model: physical and logical connection model by matrices $A$ and $C$, source flow model by matrix $S$, and related VLAN logical connection model by matrix $V$. All these matrices are defined hereafter.

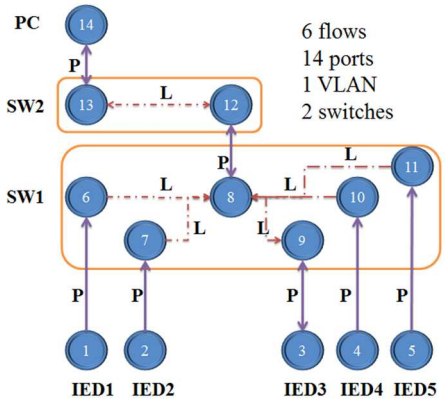



*Figure 6 Di-graph of port connections in the considered scenario.*

**Physical connection model:** If the total number of SCN ports is $P$, a $P \times P$ matrix $A$ is defined in order to describe physical connections between different ports. According to the di-graph in Figure 6, if a solid arc (physical connection) exists from port $j$ to port $i$, the related $a_{ij}$ is set to 1; if not, it is 0.

A $14 \times 14$ matrix $A$ is built as the physical connection model for this scenario involves $P = 14$ ports, in which any solid arc from port $j$ to $i$ is indicated by $a_{ij} = 1$, while the other entries are 0. There are ten physical connections, so there are 10 entries of $A$ equal to 1.

$$A = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0 \end{bmatrix}$$

**Logical connection model:** This model describes the logical connections inside the switch. A $P \times P$ matrix $C$ is defined to describe the logical connections.

In our case, dashed arcs in the di-graph of Figure 6 represent the logical connections: if there is a directed arc from $j$ to $i$, then $c_{ij} = 1$ in the $14 \times 14$ matrix $C$; otherwise, $c_{ij} = 0$. As there are eight logical connections, eight entries of $C$ are equal to 1.

$$C = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&1&1&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

## 3.3 Basic concepts of the Network Calculus

Source model and service model are built using arrival curve and service curve as basic concepts of the Network Calculus [23]. As explained in the previous section, traffic flow, as a cumulative function of time $F(\mathrm{t})$, denotes the number of bits observed in a time interval $[0, \mathrm{t}]$. MMS messages can be triggered by data changes (event-driven), also at a fixed interval (time-driven) [17]. The source model of the event-driven MMS is described as follows:

$$\alpha_k(t) = \begin{cases} 0, & for\ no\ event \\ \rho_k t + \sigma_k, & for\ event \end{cases}, \tag{10}$$

where $\rho_k$ is the data-rate MMS flow $k$, and $\sigma_k$ equals to the flow length. But, the source model of the time-driven MMS is the same as the periodic data (*i.e.*, SV messages): $\alpha(t) = \rho t + \sigma$. The upper-bound curve is identified over each flow by the arrival curve, $\alpha(t) = \rho t + \sigma$ proposed by Cruz (see Section II). Here, $F(t)$ is $(\sigma, \rho)$-upper constrained. $(\sigma, \rho)$ is thus identified by assigning a linear upper-bound curve over the measured flow $F(t)$.

On other hand, lower bounds are defined on the service provided by each port using the service curve that specifies the service provided by the service system. As introduced in Section 2, a linear $(R, T)$ widely used model (4) is considered. Now, using arrival curve, $\alpha(t)$, and service curve, $\beta(t)$, source model and service model can be constructed.

**Source model:** The port that injects a flow to the network is considered as a source port. The source model describes the properties of such flow – *e.g.*, message length, flow rate – provided by this source. Assuming $D$ as the total number of flows transmitted in the network, so a $P \times D$ matrix $S$ is defined to express the relationship between messages and their sources. In the considered scenario, the total number of flows transmitted is $D = 6$.

Matrix $S$ is involved in two parts of the algorithm: first, it is used for the distribution model of each flow from source to destination, and second, it includes the flow information passing through different ports used for the maximum delay calculation. So, for $S$ intending to be used in the flow distribution process, $s_{ij}$ denotes the length of message $j$, $g_j$, if port $i$ is the source of flow $j$. But, if $S$ is used for delay calculation, $s_{ij}$ is replaced with the $(\sigma, \rho)$-upper-bound arrival curve $\alpha_j(t) = \rho_j t + \sigma_j$ which is the related arrival curve for flow $j$. In both cases, $s_{ij}$ is set to 0 if port $i$ is not the source of flow $j$. Then, source model is built for two purposes: message distribution algorithm, and delay estimation. In order to identify parameters of $\alpha(t)$, this paper proposes an estimation method to be applied on the measured flow. First, the related linear trend is estimated as $y(t) = rt + b$ over the measured flow, where the trend slope represents $\rho$ in (2). $\rho$ means that for any time window of length $\tau$, the number of bits of the flow is limited, so peak-rate limited. Next, constant value $\sigma$ signifies the maximum number of bits that may ever be transmitted by the measured flow, so it represents the maximum length of the corresponding flow. Hence, $\sigma$ is the difference between the estimated linear trend, $y(t)$, and the flow itself, $F(t)$:

$$\sigma = max|F(t) - y(t)| \tag{11}$$

For example, Figure 7 shows the measured cumulative number of samples of the first flow generated by IED1 with the identified parameters of its corresponding arrival curve $\alpha(t)$, as $\rho = 1.98$ bps and $\sigma = 4.30$ Kbit.

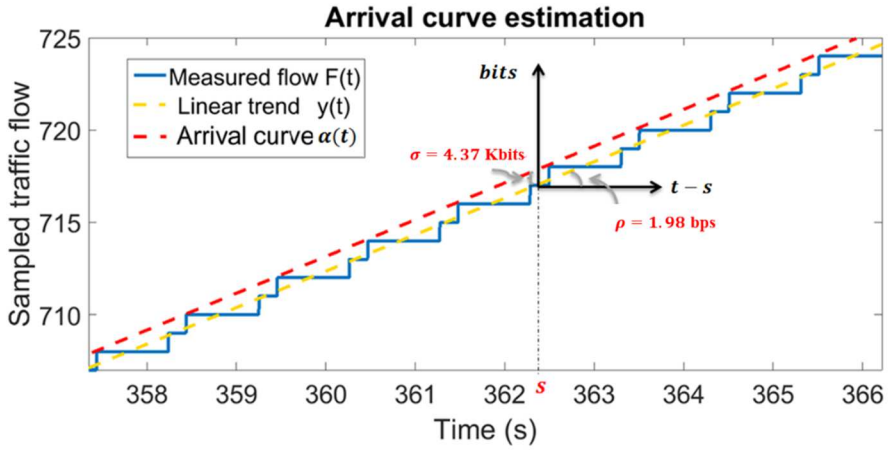



*Figure 7 Arrival curve as the upper bound for the first flow at port 1.*

Table 1 presents the identified parameters of the corresponding arrival curve for each transmitted flow.

*Table 1 Identified $(\rho, \sigma)$ parameters for all the transmitted flow.*

| | Flow1 | Flow2 | Flow3 | Flow4 | Flow5 | Flow6 |
|---|---|---|---|---|---|---|
| $\rho$ [bps] | 1.98 | 1.98 | 1.98 | 1.98 | 1.98 | 1.98 |
| $\sigma$ [Kbit] | 4.37 | 4.27 | 4.37 | 2.97 | 1.14 | 1.56 |

As it is about an identification methodology, observing – that is, measuring – sampled data is indispensable to the proposed approach. Fitting measured data on a model whose structure is known further leads to determine model parameters. In particular, determining arrival curve's parameters also works when the network's data flow rates, $\rho$, are measured first; however, full measuring of data flow is needed to also account for the flow burstiness, $\sigma$.

As the total number of flows transmitted is $D = 6$, a $14 \times 6$ matrix $S$ is defined as follows, by indicating the length of each flow ($\sigma$ in bits) generated by its source. Next, for delay estimation, all the lengths are replaced by the related arrival curves, therefore a new matrix $S$ results.

$$S = \begin{bmatrix} 4.37 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.27 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.37 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.56 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.97 & 0 & 0 \end{bmatrix}$$

$$S = \begin{bmatrix} 1.98t + 4.37 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.98t + 4.27 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.98t + 4.37 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.98t + 1.14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.98t + 1.56 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.98t + 2.97 & 0 & 0 \end{bmatrix}$$

**Service model:** Two main queue-scheduling policies can be considered in the service model: First Input First Output (FIFO), and Priority Queuing (PQ). They are used to improve the quality of service and flow congestion management. For a PQ policy, the arriving flows at a service system are tagged by a priority number from 0 to $n$, they are served from the highest priority to the lowest, and their arrival time is not important. Thus, a flow with a priority of $0 < k < n$ may be delayed by a flow with higher priority even if this latter arrives after, also by a lower-prioritized flow that is already receiving the service.

In the FIFO case, the received flows are served in order of their arrival at service systems. Here, the service policy of the validation scenario is FIFO, and each port service is $(R, T)$-lower-bounded. The estimation of the corresponding linear service curve is explained below. In case of PQ, one can refer to [17]. It is good to mention that delay estimation in case of FIFO or PQ is different in the calculation of the service rate ($R_j$) and minimum latency ($T_j$).

A new $P \times D$ matrix $K$ is built in which $k_{ij}$ denotes the corresponding service curve for flow $j$ at port $i$, $\beta_i(t) = R_i[t - T_i]^+$. The values of $R_{ij}$ and $T_{ij}$ are defined as follows:

$$R_{ij} = R_i - \sum_{q \neq j} \rho_q, \tag{12}$$

where $R_i$ (in bps) denotes the switch transmission rate as specified in the switch datasheet. The transmission rate of each port is affected by the flows passing simultaneously through the same port. So, if more than one flow pass through the same port, for serving flow $j$, the transmission rate of that port is reduced by the other flow rates, and the experienced latency is increased:

$$T_{ij} = T_i + \frac{\sum_{q \neq j} \sigma_q + T_i \times \sum_{q \neq j} \rho_q}{R_i - \sum_{q \neq j} \rho_q}, \tag{13}$$

where $T_i$ (in s) denotes the minimum latency if only one message passes through a port, and it is indicated in the switch datasheet. For more than one flow passing through the same port, latency is increased depending on the other flows lengths and transmission rates.

## 3.4 Message distribution algorithm

Prior to delay calculation, it is necessary to obtain the distribution path of each flow, since the total delay experienced by a flow is the sum of all the delays experienced while passing at each port from source to destination. This path contains physical (cabling) and logical (switch control logic) connections. First, the distribution path is separately described for each flow by a $P \times D$ matrix $S_j$.

Thus, the algorithm should be executed $D$ times, *i.e.*, where $D$ is the total number of flows – in our case, 6 items. All the basic matrices (*i.e.,* $A$, $C$, $V$, $S$) are already defined for the considered scenario. For flow $j$ passing

through a path, matrices $C_j$ and $S_j$ should be defined specifically for that flow.

The same as for ports, switches are numbered for the distribution algorithm. $k$ represents the number of switch levels where a flow enters. By setting $k = 1$, a flow distribution is modeled by (14) indicating a logical connection:

$$S_j^{3k-2} = -(C_j \times S_j^{3(k-1)}) \tag{14}$$

The message is forwarded to the next port through a physical connection:

$$S_j^{3k-1} = -(A \times S_j^{3k-2}) \tag{15}$$

Each entry has a negative or positive sign that denotes entering to the port or exiting from the port, respectively. At this point, the algorithm needs to check if the flow has arrived at the destination. To this end, all the non-zero entries of $S_j^{3k-1}$ are set to 0 if their related port $i$ is a non-switch port, and results in the corresponding matrix $S_j^{3k}$. If $S_j^{3k}$ is equal to zero, this indicates the delivery of the flow, and $n = k$ is the number of switches the message is passing through before being delivered. In case of any non-zero entry, the switch level k is incremented, $k = k + 1$, to repeat the algorithm for the same flow to calculate the message distribution through the next switch level. In our case, as there are two switches in the network, the switch level for each flow is $n = 2$. Finally, the total distribution matrix $S_j$ is obtained through the following addition for each flow $j$:

$$S_j = S_j + S_j^0 + \sum_{j=1}^{D} S_j^{3k-2} + \sum_{j=1}^{D} S_j^{3k-1}, \tag{16}$$

with $D = 6$ being the total number of flows, and $P = 14$ the total number of ports in the discussed scenario. The total message distribution model is found out as the sum of all $S_j$ matrices by a $P \times D$ matrix $S^T$ which expresses the distribution paths of all the flows:

$$S^T = \sum_{j=1}^{D} S_j \tag{17}$$

$$S^T = \begin{bmatrix} 4.37 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.27 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.37 & -2.97 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.56 \\ -4.37 & 0 & 0 & 0 & 0 & 0 \\ 0 & -4.27 & 0 & 0 & 0 & 0 \\ -4.37 & 4.27 & 4.37 & -2.97 & 1.14 & 1.56 \\ 0 & 0 & -4.37 & -2.97 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.14 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.56 \\ -4.37 & -4.27 & -4.37 & 2.97 & -1.14 & -1.56 \\ 4.37 & 4.27s & 4.37 & -2.97 & 1.14 & 1.56 \\ -4.37 & -4.27 & -4.37 & 2.97 & -1.14 & -1.56 \end{bmatrix}$$

## 3.5 Delay estimation

In this section, the message distribution model will serve for computing the experienced delay at each port for different flows. Finally, by adding all the port-delay values, the total delay is calculated for each flow. In order to do that, the source model and service model should be built up according to the related arrival curves (at each port) and service curves (for each flow). So, for each flow distribution, the proper parameters of arrival curves are identified according to the identification process explained in Subsection 3.3. Next, since matrix $S$ is to be used for delay estimation process, its entries are replaced by the proper arrival curves. So, $S^T$ is transformed to $S^D$ by being filled with arrival curves at each port. Hence, a $14 \times 6$ matrix $S^D$ is obtained:

$$S^D = \begin{bmatrix} 1.98t + 4.37 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.98t + 4.27 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.98t + 4.37 & -1.98t - 2.97 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.98t + 1.14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.98t + 1.56 \\ -1.98t - 4.37 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.98t - 4.27 & 0 & 0 & 0 & 0 \\ -1.98t - 4.37 & 1.98t + 4.27 & 1.98t + 4.37 & -1.98t - 2.97 & 1.98t + 1.14 & 1.98t + 1.56 \\ 0 & 0 & -1.98t - 4.37 & 1.98t + 2.97 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.98t - 1.14 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.98t - 1.56 \\ -1.98t - 4.37 & -1.98t - 4.27 & -1.98t - 4.37 & 1.98t + 2.97 & -1.98t - 1.14 & -1.98t - 1.56 \\ 1.98t + 4.37 & 1.98t + 4.27 & 1.98t + 4.37 & -1.98t - 2.97 & 1.98t + 1.14 & 1.98t + 1.56 \\ -1.98t - 4.37 & -1.98t - 4.27 & -1.98t - 4.37 & 1.98t + 2.97 & -1.98t - 1.14 & -1.98t - 1.56 \end{bmatrix}$$

Service model in the form of a $D \times P$ matrix $K$ is built based on $S^D$ by replacing all the arrival curves with the corresponding service curves of each port through which a flow passes. As discussed in Section 2, $R_{ij}$ and $T_{ij}$ are calculated while considering all the flows passing through the same port. If there is only one flow passing through a port, $R$ (port transmission rate) and $T$ (minimum latency of the port) are taken from the switch datasheet. But if there are more than one flow, as the switch policy in this experiment is FIFO, $R_{ij}$ and $T_{ij}$ are obtained using (12) and (13). The service model for the considered scenario is obtained by replacing the service curves assigned to

each flow passing through a port. So, in the considered case the service model is built as a $14 \times 6$ matrix $K$:

$$K = \begin{bmatrix} (2.6,2.2) & 0 & 0 & 0 & 0 & 0 \\ 0 & (2.6,2.2) & 0 & 0 & 0 & 0 \\ 0 & 0 & (2.6,2.2) & (2.6,2.2) & 0 & 0 \\ 0 & 0 & 0 & 0 & (2.6,2.2) & 0 \\ 0 & 0 & 0 & 0 & 0 & (2.6,2.2) \\ (2.6,2.2) & 0 & 0 & 0 & 0 & 0 \\ 0 & (2.6,2.2) & 0 & 0 & 0 & 0 \\ (1.45,4.97) & (1.44,4.97) & (1.45,4.97) & (2.43,3.36) & (1.05,8.15) & (1.11,7.99) \\ 0 & 0 & (2.53,4.) & (2.58,4.94) & 0 & 0 \\ 0 & 0 & 0 & 0 & (2.6,2.2) & 0 \\ 0 & 0 & 0 & 0 & 0 & (2.6,2.2) \\ (2.91,5.39) & (2.89,5.49) & (2.92,5.29) & (2.79,2.37) & (2.46,4.30) & (2.53,4.23) \\ (2.91,5.39) & (2.89,5.49) & (2.92,5.29) & (2.79,2.37) & (2.46,4.30) & (2.53,4.23) \\ (4.4,0) & (4.4,0) & (4.4,0) & (4.4,0) & (4.4,0) & (4.4,0) \end{bmatrix}$$

In this scenario all the ports of the network – switch ports, IEDs' ports, and SCADA's ports – are assumed to have the same service values, *i.e.*, $R$ and $T$.

Now, confirming the min-plus algebra [27], it is possible to calculate the maximum message transmission delay at each port using the information of $S^D$ and $K$. The output flow bounds of a service system can be computed from the min-plus deconvolution of arrival curve and system service curve as explained by (7) and delay of a flow is bounded by (8). Now, the maximum delay at each port with an arrival curve $\alpha(t)$ of $(\rho, \sigma)$-form and a service curve $\beta(t)$ of $(R, T)$-form is upper-bounded by (9) as: $D_{ij}(t) \leq h(\alpha_i, \beta_j) = T_i + \frac{\sigma_j}{R_i}$. Following (9), delay values at each port are estimated in a form of a $D \times P$ matrix $Y$. Related values of $R$, $T$, and $\sigma$ are available in the corresponding entry of $S^D$ ($S_{ij}^D$) and $K$ ($K_{ij}$) matrices. Therefore, in our case, estimated delay for each port is shown in Table 2.

*Table 2 Maximum estimated delay (μs) for each flow passing through different ports for the considered load-shedding scenario.*

| Port number | Flow1 | Flow2 | Flow3 | Flow4 | Flow5 | Flow6 |
|---|---|---|---|---|---|---|
| 1 | 3.88 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 3.84 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 3.88 | 3.34 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 2.64 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 2.80 |
| 6 | 3.88 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 3.84 | 0 | 0 | 0 | 0 |
| 8 | 7.98 | 7.94 | 7.98 | 4.59 | 9.24 | 9.40 |
| 9 | 0 | 0 | 4.68 | 4.09 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 2.64 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 2.80 |
| 12 | 6.89 | 6.97 | 6.79 | 3.43 | 4.76 | 4.85 |
| 13 | 6.89 | 6.97 | 6.79 | 3.43 | 4.76 | 4.85 |
| 14 | 0.99 | 0.97 | 0.99 | 0.68 | 0.26 | 0.35 |

# 4 Synthesis and proposed methodology for delay estimation

Figure 8 represents the flowchart of the proposed methodology for delay evaluation that integrates an application of the considered analytical model.

## 4.1 Flowchart of the proposed methodology

The analytical model is applied in this order: first, the message distribution model is constructed from source to destination; second, a source-flow model and a service-flow model are built up to characterize each flow while it is served to pass through different ports. Finally, the maximum message transmission delay is calculated based on the corresponding flow and service information at each port.

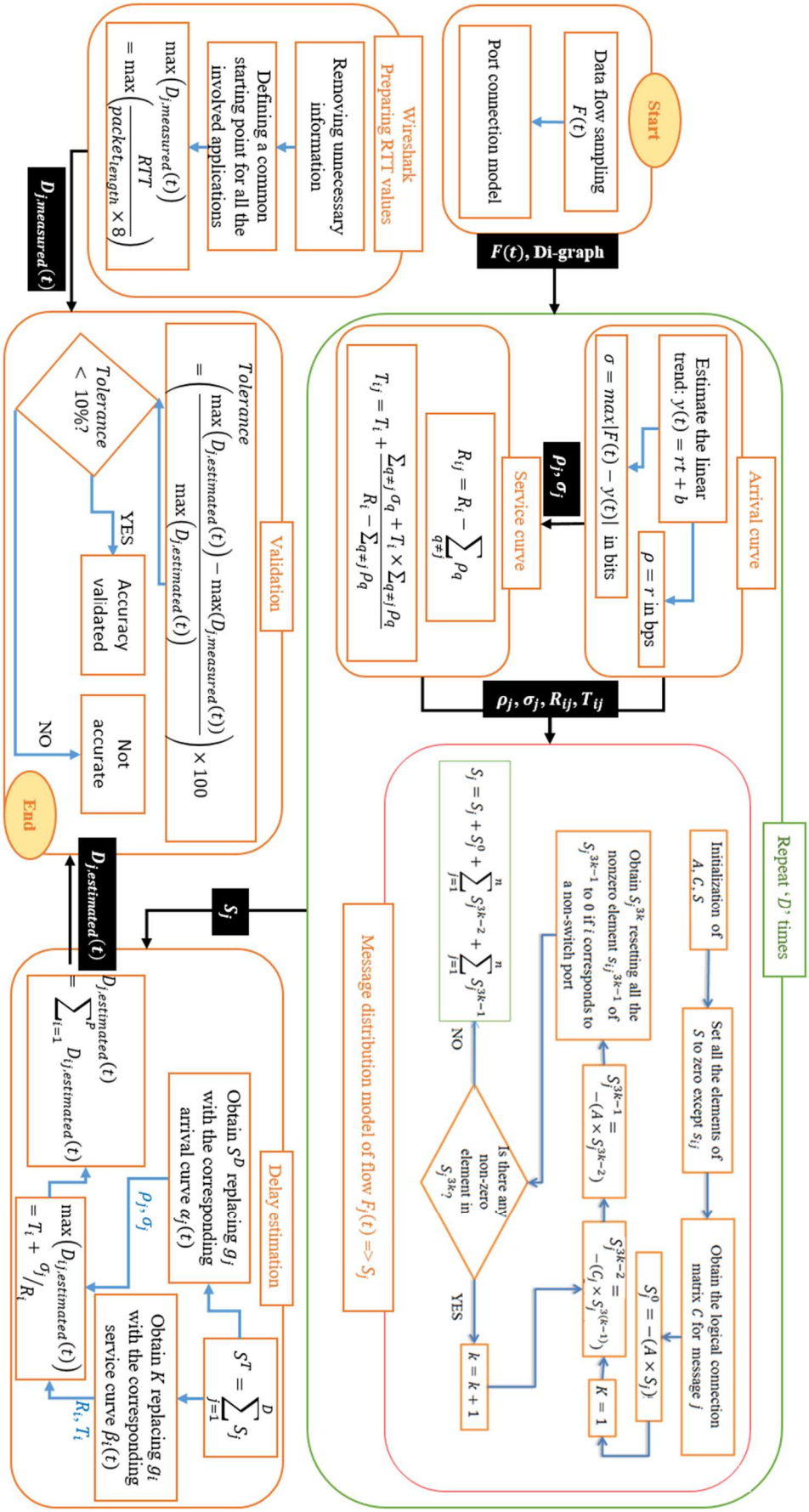
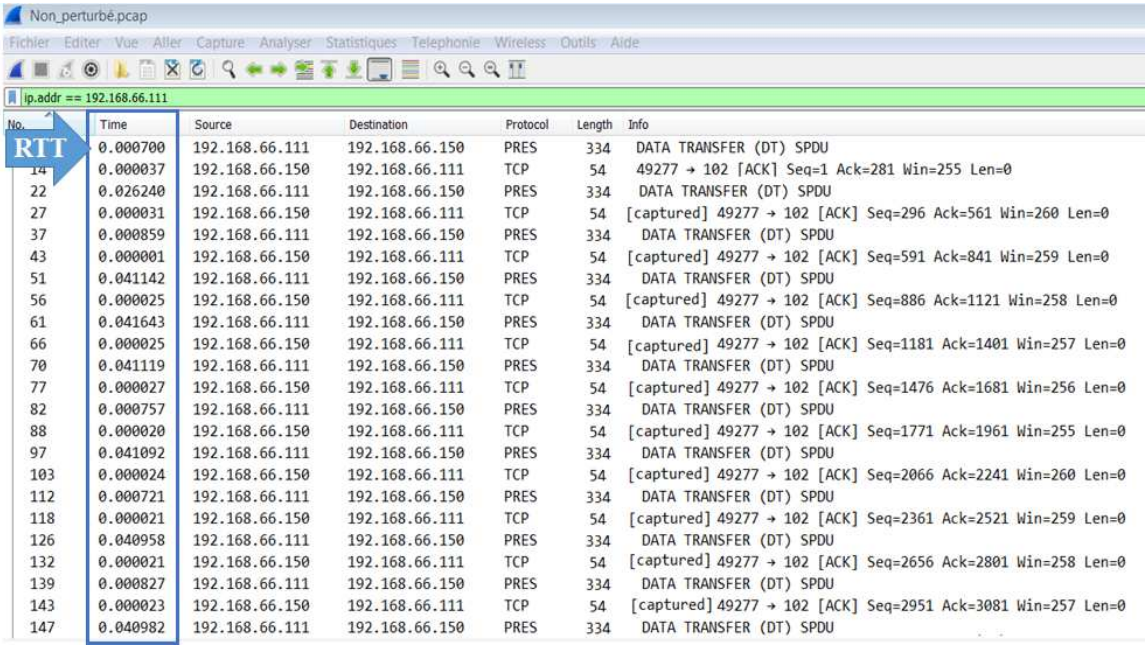



*Figure 8 The proposed comprehensive methodology for the maximum message transmission delay is detailed step by step using the represented flowchart.*

## 4.2 Estimated delay against measured values

Now, the estimation results are validated through a comparison performed against the measured values of the Wireshark network analyzer. As an example, Figure 9 illustrates the conversation of the first flow and SCADA. Two sides of the conversation are mentioned by different IP addresses (source and destination columns).

As the raw information, Wireshark measures the Round Trip Time (RTT) indicated on the RTT column of Figure 9, which is the transmission time for a packet to be surely delivered, including acknowledgment. So, RTT is the time for a packet of data (DT in bytes) from sending to receiving its acknowledgment (ACK) back. To make RTT comparable to the previously estimated delay, data flows monitored by Wireshark need to be preprocessed. To do so, the conversations of IEDs and SCADA are separated, only DT packets of the sender and the ACK of receiver should be included.




| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| | 0.000700 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 14 | 0.000037 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | 49277 → 102 [ACK] Seq=1 Ack=281 Win=255 Len=0 |
| 22 | 0.026240 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 27 | 0.000031 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=296 Ack=561 Win=260 Len=0 |
| 37 | 0.000859 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 43 | 0.000001 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=591 Ack=841 Win=259 Len=0 |
| 51 | 0.041142 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 56 | 0.000025 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=886 Ack=1121 Win=258 Len=0 |
| 61 | 0.041643 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 66 | 0.000025 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=1181 Ack=1401 Win=257 Len=0 |
| 70 | 0.041119 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 77 | 0.000027 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=1476 Ack=1681 Win=256 Len=0 |
| 82 | 0.000757 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 88 | 0.000020 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=1771 Ack=1961 Win=255 Len=0 |
| 97 | 0.041092 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 103 | 0.000024 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=2066 Ack=2241 Win=260 Len=0 |
| 112 | 0.000721 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 118 | 0.000021 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=2361 Ack=2521 Win=259 Len=0 |
| 126 | 0.040958 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 132 | 0.000021 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=2656 Ack=2801 Win=258 Len=0 |
| 139 | 0.000827 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |
| 143 | 0.000023 | 192.168.66.150 | 192.168.66.111 | TCP | 54 | [captured] 49277 → 102 [ACK] Seq=2951 Ack=3081 Win=257 Len=0 |
| 147 | 0.040982 | 192.168.66.111 | 192.168.66.150 | PRES | 334 | DATA TRANSFER (DT) SPDU |

*Figure 9 The monitored conversion of the first flow while IED3 sends the measurements to SCADA. The time column indicates the measured RTT values by Wireshark.*

Unnecessary information is removed from the conversation, such as Media Address Control (MAC) management packets, and DT packets sent by the receiver, which do not contain any information related to our communication scenario.

Delay estimation is done per bit, while in Wireshark it is done per packet in bytes. In addition, to obtain the maximum delay, the maximum RTT is extracted from all the measured values for each flow. The proper RTT for each flow $j$ is thus given by:

$$\max\left(D_{j,measured}(t)\right) = \max\left(\frac{RTT}{packet_{length}\times 8}\right), \tag{18}$$

where $packet_{length}$ is the sum of DT length and ACK length. The measured RTT values are compared to the estimated delay for the reason that the considered delay is composed of the time to send a message and the time to receive the acknowledgment.

Before the scenario execution, there was a peak of RTT value which affected $\max\left(D_{j,measured}(t)\right)$, and resulted in a significant difference between measured and estimated delay. Thus, a common point needs to be identified, starting from which all the three involved applications – SCADA, Wireshark, and Premium automate – perform synchronously. As they are activated manually, it may be difficult to start all three at the same time. Hence, a preprocessing phase is needed to remove the initial "synchronization" phase, a common start point is defined for all the sampled flows by shifting our analysis windows accordingly on the time axis.

The preprocessing phase ensures that the measured value to be correctly extracted and compared to the total estimated delay for each flow. Total estimated delay is obtained for each flow by a vertical sum of the columns in $Y$. Related values of estimated and measured delay are compared in Table 3.

*Table 3 The estimated delay in µs for the perturbed scenario is compared to the measured values using Wireshark network analyzer.*

| | **Flow 1** | **Flow 2** | **Flow 3** | **Flow 4** | **Flow 5** | **Flow 6** |
|---|---|---|---|---|---|---|
| **Estimated delay** | 30.5 | 30.5 | 31.1 | 19.6 | 24.3 | 25.1 |
| **Measured delay** | 29.9 | 29.7 | 30.0 | 18.5 | 22.8 | 23.8 |
| **Tolerance** | 2.20% | 2.62% | 3.76% | 5.65% | 6.11% | 5.18% |
| **Exact difference** | 0.67 | 0.80 | 1.17 | 1.11 | 1.49 | 1.30 |

As validation criterion, (19) gives the tolerance (error value in percentage) for each flow $j$, the exact values being calculated as the simple differences of the measured and estimated delay.

$$Tolerance = \left( \frac{\max\left(D_{j,estimated}(t)\right) - \max(D_{j,measured}(t))}{\max\left(D_{j,estimated}(t)\right)} \right) \times 100 \quad (19)$$

All tolerance values in Table 3 are less than $10\%$, which indicates a good quality of estimation. In addition, all estimated values are greater than the measured values, which is typical as it is about an upper bound of the delay.

# 5 Conclusion

This paper proposes a methodology of evaluation for the maximum message transmission delay using an analytical model based on Network Calculus Theorem within a communication network in smart grids over a real-time IEC 61850-traffic generator. The arrival curve is fitted over each flow to assign an upper bound. To fit the arrival curve -- $(\rho, \sigma)$-model -- over the flows, an identification methodology is provided in this paper. The service curve is determined based on the port properties to find the corresponding lower bound. Then, the maximum message transmission delay is calculated according to the flow-distribution model. The validation methodology is performed on a real-time scenario that includes three communicating IEDs participating in a common task of load shedding, while two other IEDs are added to generate background, perturbation traffic. To evaluate the estimation effectiveness, estimated values are compared to the measured delay by a network analyzer, Wireshark. The comparison shows very satisfactory estimation errors (within $10\%$).

Therefore, the maximum delay estimation using the analytical method applied on an intelligent substation gives a good estimation, but it is limited to the cases with a completely known network structure, which is the case of the proprietary networks. If *a priori* knowledge about the communication network is not available, some other estimation approach should be envisaged.

Availability of a method of maximum communication delay estimation is of particular importance in any smart grid application involving closed-loop control, as negative effects of improperly large feedback delays upon closed-loop stability are widely known. Thus, a priori knowledge of maximum delay is significant for deriving pertinent restrictions - of communication network initial configuration and/or operation - to apply in order to avoid global instability.

***Recommendations for future work***

The method is applied on pre-sampled traffic data*, i.e.*, off-line. On-line application of this method on a sliding $T$-width time window on an a-priori-not-known, possibly evolving network, would give a time evolution of the maximum delay. Such information may further be used with machine-learning algorithms to predict delay upper bounds. In addition, more complex scenarios such as inter-substation communication with faults, modern QoS strategies (*e.g.*, Priority Queuing) and dynamic networks (*e.g.*, delay optimal routing) or some message characteristics such as variable-length data, on-line flow studies, while they are generated by different sources with different flow rates, are interesting to be analyzed.

Being based on effective measurements of data flows, the proposed identification methodology appears to be easy to extend for cases where more complex functions – such as, for example, congestion control – are embedded into the service units (switches or routers), as long as such functions help preserving regularity of the communication flow, a basic assumption of the presented method.

Another interesting generalization direction regards the case where stochastic – instead of linear – flow rates characterize globally the communication flow. Whereas relaxing this assumption appears to be more realistic, generalization does not seem difficult as long as flow rates' stochasticity remains within some defined bounds. Indeed, if the initially assumed flow regularity is preserved, one can reasonably consider that the actual flow "lies" between two flows, characterized by the two bounds, respectively. Hence, the maximum delay estimation methodology can be applied to each of these flows, and then the maximum among these estimated maxima would represent the worst case of delay.

# References


[1] Müller, S., C., Georg, H., Nutaro, J., J.,Widl, E., Deng, Y., Palensky, P., Awais, M. U., Chenine, M, Küch, M. and Stifter, M., "Interfacing power system and ICT simulators: Challenges, state-of-the-art, and case studies," *IEEE Transactions on smart grid,* vol. 9, no. 1, pp. 14-24, Jan 2018.

[2] Gungor, V., C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., Hancke, G., P., "A Survey on smart grid potential applications and communication requirements," *IEEE Transactions on Industrial Informatics,* vol. 9, no. 1, pp. 28-42, Feb 2013.

[3] Qureshi, T., Alvi, R., "Substation Automation System Design Using IEC 61850 Communication Protocol," *I-Manager's Journal on Instrumentation & Control Engineering,* vol. 6, no. 2, pp. 21-30, 2018.

[4] Hao, W., Yang, Q., "Data traffic characterization in intelligent electric substations using FARIMA based threshold model," *Energy Procedia,* vol. 145, pp. 413-420, July 2018.

[5] Feizimirkhani, R., Bratcu, A. I., and Besanger, Y., "Time-series Modelling of IEC 61850 GOOSE Communication Traffic between IEDs in smart grids - a parametric analysis," *IFAC-PapersOnLine, ScienceDirect, ELSEVIER,* vol. 51, no. 28, pp. 444-449, Dec. 2018.

[6] Yang, T., Zhao, R., Zhang, W. and Yang, Q., "On the modeling and analysis of communication traffic in intelligent electric substation," *IEEE Transactions on Power Delivery,* vol. 32, no. 3, pp. 1329-1338, June 2017.

[7] Thomas, M. S., and Ali, I., "Reliable, fast, and deterministic Substation Communication Network architecture and its performance simulation," *IEEE Transactions on Power Delivery,* vol. 25, no. 4, pp. 2364-2370, Oct 2010.

[8] S. Kumar, N. Das, J. Muigai and S. Islam, "Performance evaluation of data transmission in a single and double bus network within the utility substation based on IEC 61850," *IEEE PES General Meeting | Conference & Exposition,* pp. 1-5, 30 October 2014.

[9] Sidhu, T., S., and Yin, Y., "Modeling and simulation for performance evaluation of IEC61850-based substation communication systems," *IEEE transactions on Power Delivery,* vol. 22, no. 3, pp. 1482-1489, July 2007.

[10] Kanabar, P. M., Kanabar, M. G., El-Khattam, W., Sidhu, T. S., and Shami, A., "Evaluation of communication technologies for IEC 61850 based distribution automation system with distributed energy resources," *IEEE Power & Energy Society General Meeting,* pp. 1-8, July 2009.

[11] Konka, J. W., Arthur, C. M., Garcia, F. J., and Atkinson, R. C., "Traffic generation of IEC 61850 sampled values," *IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS),* pp. 43-48, 2011.

[12] Georg, H., Dorsch, N., Putzke, M., and Wietfeld, C., "Performance evaluation of time-critical communication networks for smart grids based on IEC 61850," *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS),* pp. 43-48, 2013.

[13] Huang, C., Li, F., Ding, T., Jiang, Y., Guo, J., and Liu, Y., "A Bounded Model of the Communication Delay for System Integrity Protection Schemes," *IEEE Transactions on Power Delivery,* vol. 31, pp. 1921-1933, Feb 2016.

[14] Yang, H., Cheng, L., and Ma, X., "Analyzing Worst-Case Delay Performance of IEC 61850-9-2 Process Bus Networks Using Measurements and Network Calculus," *The Eighth International Conference on Future Energy Systems,* p. 12–22, May 2017.

[15] Dos Santos, A., Soares, B., Fan, C., Kuipers, M., Sabino, S., Grilo, A. M. R. C., Pereira, P. R. B. A., Nunes, M. S., and Casaca, A., "Characterization of Substation Process Bus Network Delays," *IEEE Transactions on Industrial Informatics,* vol. 14, no. 5, pp. 2085-2094, May 2018.

[16] Docquier, T., Song, Y. Q., Chevrier, V., Pontnau, L., and Ahmed-Nacer, A., "IEC 61850 over TSN: traffic mapping and delay analysis of GOOSE traffic," *25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA),* pp. 246-253, 2020.

[17] Zhang, Y., Cai, Z., Li, X., and He, R., "Analytical Modeling of Traffic Flow in the Substation Communication Network," *IEEE Transactions on Power Delivery,* vol. 30, no. 5, pp. 2119-2127, Oct 2015.

[18] Fidler, M., and Schmitt, J. B., " On the way to a distributed systems calculus: An end-to-end network calculus with data scaling," *Proceedings of the joint international conference on Measurement and modeling of computer systems ,* vol. 34, no. 1, pp. 287-298, June 2006.

[19] Le Boudec, J. Y., and Thiran, P., Network Calculus – a Theory of Deterministic Queuing Systems for the Internet, 1st ed. ed., vol. 2050, S. B. Heidelberg, Ed., 2004, pp. 83-157.

[20] 61850-9-3, IEC/IEEE, "Communication networks and systems for power utility automation – Part 9-3: Precision time protocol profile for power utility automation," *Standard,* 2016.

[21] Ingram, D. M. E., Steinhauser, F., Marinescu, C., Taylor, R. R., Schaub, P., and Campbell, D. A., "Direct evaluation of IEC 61850-9-2 process bus network performance," *IEEE Transactions on Smart Grid,* vol. 3, no. 4, pp. 1853-1854, Dec 2012.

[22] Bondorf, S., and Geyer, F., "Generalizing Network Calculus Analysis to Derive Performance Guarantees for Multicast Flows," *10th EAI International Conference on Performance Evaluation Methodologies and Tools ,* no. 8, pp. 2-9, May 2017.

[23] Cruz, R. L., "A calculus for network delay, part I: Network elements in isolation," *IEEE Transactions on Information Theory,* vol. 37, no. 1, pp. 114-131, Jan 1991.

[24] Cruz, R. L., "A calculus for network delay, part Ⅱ: Network analysis," *IEEE Transactions on Information theory,* vol. 37, no. 1, pp. 132-141, 1991.

[25] Georges, J. P., Krommenacker, N., Divoux, T., and Rondeau, E., "A design process of switched Ethernet architectures according to real-time application constraints," *Engineering Applications of Artificial Intelligence,* vol. 19, pp. 335-344, Apr 2006.

[26] Bouillard, A., Boyer, M., and Le Corronc, E., Deterministic network calculus: from theory to practical implementation, 1 st ed. ed., Wiley-ISTE, Ed., 2018.

[27] Liebeherr, J., "Duality of the max-plus and min-plus network calculus," *Foundations and Trends in Networking,* vol. 11, no. 3-4, pp. 139-282, July 2017.

[28] B. A. Forouzan, Data communications and networking, McGraw-Hill, 2012.