IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 5, 2019, accepted August 25, 2019, date of publication September 4, 2019, date of current version September 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939497

Multi-Level Time-Sensitive Networking (TSN) Using the Data Distribution Services (DDS) for Synchronized Three-Phase Measurement Data Transfer

TANUSHREE AGARWAL[1], (Student Member, IEEE),
PAYAM NIKNEJAD[1], (Student Member, IEEE),
M. R. BARZEGARAN[1], (Member, IEEE),
AND LUIGI VANFRETTI[2], (Student Member, IEEE)
[1]Department of Electrical Engineering, Lamar University, Beaumont, TX 77710, USA
[2]Department of Electrical and Computer Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Corresponding author: Tanushree Agarwal (tagarwal@lamar.edu)

This work was supported in part by the Lamar University, and in part by National Instruments.

ABSTRACT This paper presents the design and implementation of a Multi-level Time Sensitive Networking (TSN) protocol based on a real-time communication platform utilizing Data Distribution Service (DDS) middleware for data transfer of synchronous three phase measurement data. To transfer ultra-high three phase measurement samples, the DDS open-source protocol is exploited to shape the network's data traffic according to specific Quality of Service (QoS) profiles, leading to low packet loss and low latency by synchronizing and prioritizing the data in the network. Meanwhile the TSN protocol enables time-synchronization of the measured data by providing a common time reference to all the measurement devices in the network, making the system less expensive, more secure and enabling time-synchronization where acquiring GPS signals is a challenge. A software library was developed and used as a central Quality of Service (QoS) profile for the TSN implementation. The proposed design and implemented real-time simulation prototype presented in this paper takes in consideration diverse scenarios at multiple levels of prioritization including publishers, subscribers, and data packets. This allows granular control and monitoring of the data for traffic shaping, scheduling, and prioritization. The major strength of this protocol lies in the fact that it's not only in real time but it's time-critical too. The simulation prototype implementation was performed using the Real Time Innovation (RTI) Connext connectivity framework, custom-built MATLAB classes and DDS Simulink blocks. Simulation results show that the proposed protocol achieves low latency and high throughput, which makes it a desired option for various communication systems involved in microgrids, smart cities, military applications and potentially other time-critical applications, where GPS signals become vulnerable and data transfer needs to be prioritized.

INDEX TERMS Data distribution services, latency, multi-level, network-shaping, prioritization, quality of service (QoS), time-sensitive network (TSN), throughput, three-phase measurements, power system monitoring.

I. INTRODUCTION

A. MOTIVATION

With the current fast paced development of monitoring technologies such as sensors, embedded systems and cloud computing, a new generation of integrated measurement and monitoring systems have emerged; with some of which being low cost, low power, they have become an indispensable part of our daily lives. Many such devices are being used in large numbers (i.e. 1000 devices/person), and they form Internet of Things (IoT) [1]–[3]. IoT has applications in numerous areas including home automation, industrial automation, mobile healthcare, elderly assistance, intelligent energy management, smart grids, automotive, and data traffic

---

The associate editor coordinating the review of this manuscript and approving it for publication was Long Wang.

management [4], [5], to name a few. Hence, the evolving IoT consists of varied sensors, electronic devices, communication technologies, software applications, and ever-growing data in multiple formats, bringing a tremendous challenge for inter-communications interoperability at different levels.

To take advantage of these new technologies in the power industry and address existing challenges with open architectures and interoperability in mind, middleware offers an attractive approach for data transfer and to ease application development. It provides common services for applications integrating heterogeneous computing and communications devices and supporting interoperability within the diverse applications and services running on these devices. To help the headway of IoT middleware, numerous working frameworks have been created [6]–[9]. In this paper, the Time Sensitive Networking (TSN) and Distributed Data Service protocols form the basis for a new approach that can be used for data transfer of ultra-high three-phase sampled data with Quality of Service Guarantees (QoS), these protocols are reviewed in the following sections.

B. LITERATURE REVIEW

Even though middleware eliminates the complexities of the underlying system architecture and emphasizes on solving the task at hand [10], [11], there are additional challenges that come with network growth, as it becomes overwhelmed with additional devices and their related data interactions; leading to data loss, network traffic and bottle necks. To address this challenge, an ideal and structured middleware should comply with the following requirements:

- Provide a common time reference that can work in conjunction with or replacement of GPS technology, which has been shown to be vulnerable to impairment (e.g. spoofing attacks).
- Guarantee time critical communication in real-time by providing latency and throughput guarantees
- Decrease or maintain the existing computational burden without compromising on amount of data and its security by prioritizing data transfer
- Allow for prioritization levels in the network, and
- Be available in an open design to facilitate interoperability.

A middleware meeting such requirements could be exploited in Wireless Sensor Networks (WSN's), Radio Frequency Identification (RFID), Machine-to-Machine communications (M2M), Supervisory Control and Data Acquisition (SCADA), or even in a combination of these systems [12]–[14]. Such kind of combinations lead to a great number of design approaches, including message-centric, data-centric, event-based, service-oriented, Virtual Machine (VM)-based, agent-based, Tuple-spaces, database-oriented, application-specific [15], [16], to name a few.

The message-centric approach defines a set of messages and data formats, while the middleware has no information about message content or data types. Hence, each receiver is responsible for assuring the correctness of data delivered [18]. As a result, each receiver undergoes integrity checks and message parsing leading to a complex and time-consuming control application layer, poor network utilization, wasted bandwidth, and overall limited system flexibility. In a data-centric approach, the message is built by the middleware instead to report or update a change in the system state. As the message is generated by the middleware itself, it is aware of the type and content of the message and the data-centric infrastructure performs filtering, message parsing and all required integrity checks to ensure the correctness of the received data. Changing the message processing responsibility from the application layer to middleware increases the reliability and flexibility and reduces the errors. It also improves network utilization and optimum use of network bandwidth [19]. Finally, in an event-based middleware, components, applications, and all the other participants interact through events. Events are propagated from sending application components (i.e. publishers) to receiving application components (i.e. subscribers [17]). In this design approach, the communication protocol can be refined as either a message-centric and data-centric protocol.

DDS is an open communication protocol that is used to implement M2M learning at the edge where two machines may be from different vendors. It supports characteristics such as message parsing, data filtering, error checking, and network provides a network description. This framework is a real-time system that is based on the publisher-subscriber concept and supports peer to peer communication. Applying QoS by DDS distinguishes the proposed protocol from other communication methods. Different QoSs can be applied to each data type, making the framework more agile and adaptive While DDS has so much to offer, there is a need for a technology that uses this platform in such a way that the models generated on it have a sense of time, prioritization, and synchronization. These capabilities are provided using the new (TSN technology. While previous work has highlighted the value of using similar communication approaches similar to TSN [23], there are currently no fully developed or operated networks with TSN technology as its core.

C. PAPER CONTRIBUTIONS

The main contribution of this paper is to propose a new approach for multi-level TSN-based real-time communications utilizing the DDS for synchronized three-phase measurement data transfer. The specific contributions of this work are to (a) propose, (b) design, (c) build a simulation prototype and (d) demonstrate using real-time simulation the effectiveness of the proposed communications platform. These contributions are substantiated with simulation experiments that aim to show:

- That it is possible to provide a common time reference for all devices without additional hardware, potentially replacing (or complementing) GPS technology by using TSN via DDS.

- That using TSN via DDS it is possible to provide time critical communications in real time. i.e. providing data packets with a specified priority level.
- That is possible to support higher data rates and faster measurements that currently existing synchronized measurement technology (i.e. Phasor Measurement Unit-based systems) by using the proposed TSN via DDS approach.
- That is possible to provide high throughput and low latency through QoS profile design, i.e. that is, to achieve these metrics by specifying the priority at all levels (publishers, subscribers, and data) when enabling TSN using DDS as proposed in this paper.

The simulation prototype of the communication platform and data transfer approach proposed in this paper have been implemented within a simulation tool and used for the design and prediction of network traffic and node priorities using an event-based data-centric approach. Simulation tests have been carried out for data transfer using real-life wind farm measurements where high throughput is necessary to transport data with high resolution information, while monitoring requirements are guaranteed though real-time and time-critical communications.

The remainder of the paper is organized as follows: Section II describes TSN, its standards, capabilities and its characteristics. Section III describes the simulation prototype implementation including the simulation model and the developed software library, in detail. Section IV illustrates the application of the proposed approach by replaying high-resolution measurements from a real-life wind turbine whose measurement devices do not yet provide the functionalities proposed herein. Section V summarizes and concludes with the major findings of this work.

II. TIME SENSITIVE NETWORKING – A PRIMER FOR THREE-PHASE POWER SYSTEM DATA TRANSFER

TSN is the advancement of standard Ethernet, particularly the IEEE 802.1AS standard, that suggests time synchronization of devices utilizing packet transfer over Ethernet. This facilitates traffic scheduling and system configuration enabling deterministic communication over Ethernet by allowing users to schedule time-critical data across a network [20], [21]. IEEE 802.1AS is an IEEE 1588 profile that provides a common time reference to all the nodes within the IEEE 802.1AS subnet. It synchronizes multiple devices using packet-based communication and makes it possible over long distances without any signal propagation delay impact. Input/output (I/O) synchronization on devices using this profile is less than 1 μs. It can be further reduced to the hundreds of nanoseconds range, depending on the configuration of the system.

TSN provides a new option and an alternative to GPS-based synchronization while simultaneously being part of the connectivity (network) of the system. Being based on an open standard i.e. continuous evolution, TSN may be implemented in diverse ways and may have different meanings for different applications. TSN includes the capabilities such as time synchronization, traffic scheduling, system configuration, and priority scheduling.

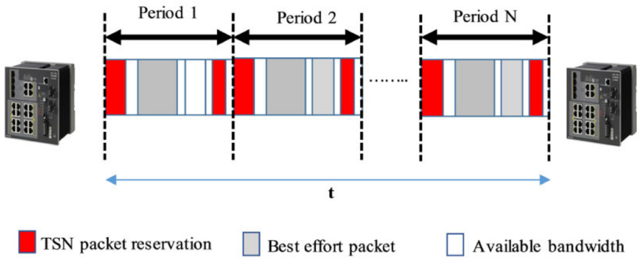


FIGURE 1. A data packet being transmitted between two TSN-enabled devices.

Time synchronization means all devices including network components share a common time reference and can synchronize with each other by synchronizing internal time signals with respect to that reference.

Traffic Scheduling provides mechanisms to ensure that information is delivered with a certain level of determinism for real-time communication without disrupting the currently existing prioritization mechanisms of non-TSN Ethernet. Because all devices are aware of what time it is and are synchronized, communication traffic can be deterministically scheduled based on priority, as opposed to the nondeterministic (but fast) traffic in previous standards.

System configuration is to standardize the parameters for configuration such as reservation of communication paths, time slots, and bandwidth to handle things like fault-tolerance and mission-critical information transfer.

TSN uses time to schedule priority traffic among different end devices and switches with a shared notion of time. Each transmission link has a schedule that includes flow IDs, transmission offsets and expected payloads. Fig.1 shows data packet being transmitted between two TSN-enabled devices. In this figure there are N periods, that are designed to separate the communication on the network into fixed length, repeating time cycles. Within each period, three different time slices are configured. Time slice 1 presented in red is TSN packet reservation i.e. high priority data. While grey, time slice 2 is the best effort packet data. Therefore, a guard band that is shown by white and called available bandwidth is placed between the two time slices for a smooth transition and to protect critical data stream.

By adding the TSN capabilities to any network, synchronization can be obtained. Synchronization, at its most basic level, is ensuring that components of a system derive their acquisition or generation timing from the same source so that data taken throughout the test can be properly correlated and analyzed. Without proper synchronization, there is no way to know if two measurements were recorded simultaneously or, it is the case of stimulus/response to testing. TSN provides the basic infrastructure for synchronization and a common time reference to all devices in the system while also providing traffic scheduling and system management capabilities [22], [23].

All these capabilities are important in any system where data exchange between devices must be correctly time-stamped and arrive at its destination within a specific timeframe with minimum jitter.

III. ENABLING TSN THROUGH QOS PROFILE DESIGN AND SIMULATION MODEL IMPLEMENTATION

This section explains the basic communication model and TSN model based that is built on top of it. The communication infrastructure for a given network should be flexible and open so that it can exchange information in real-time and accommodate different devices varied in operation and vendors at present and have capabilities to extend itself in future. The sections presented below describes the basic layers of the communication model, need of the TSN model [24], its features and its implementation details.

A. LAYERS OF COMMUNICATION MODEL

The basic communication model has four basic layers hardware, software, data and function layer as shown in three-dimensional Fig. 2. Hardware layer includes the basic components like computers, data acquisition and measurement devices, sensors etc., which when combined comprise of senders and receivers in the network. Software layer binds the hardware layer components by communication protocols and technologies. Another aspect of this layer is to assign appropriate information for data exchange such as synchronization, accuracy and transfer rate. The third is the data layer, which describes the information being exchanged between the components of hardware setup and data model standards. The Data Model standard implies all the existing data models are made available and any future development, will follow the same policies and procedures. Thus it is able to provide information that a device needs for the implementation of a particular case scenario; therefore helping in selecting the proper software to be installed for the hardware components. The fourth layer is the function layer that defines the domain or zone of operation for the complete model. In addition, it also provides function-to-component mapping that helps in defining software and hardware requirements of the components.

B. TSN MODEL DESIGN

There are many communication networks based on the above stated layers out of which peer-to-peer communication is utilized here. Layers are in direct contact with each other without the third-party intervention. However, this becomes a challenge in a network that must meet real-time requirements. To overcome this problem, DDS middleware is an ideal solution as it allows for a data-centric approach to be used to meet real-time requirements.

As mentioned previously, the TSN model is constructed based on the basic layers of communication model to analyze and visualize in a technology-neutral manner. Figure 2 depicts the communication model for TSN which is established by merging the concept of interoperability layers with TSN concept. As the figure shows the TSN model consists of four interoperability layers representing function layer, data exchange and data models, communication protocol and hardware.

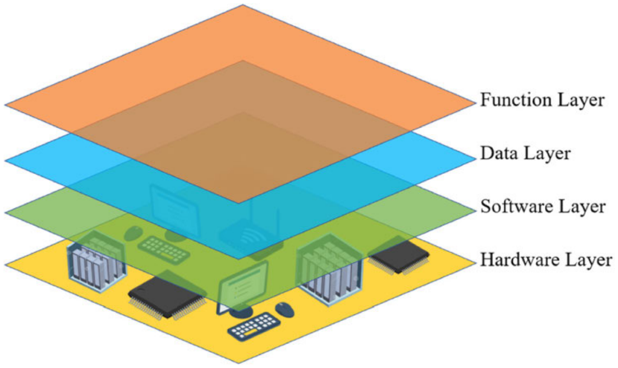


FIGURE 2. The layers of the communication model.

The proposed implementation uses the DDS existing middleware to support TSN. Because DDS is a mature standard, there are several implementations available. In this paper, the RTI DDS Simulink Blocks and MATLAB classes, along with the RTI Connext DDS connectivity framework are used. Fig. 3 shows the model defined by using DDS blocks in MATLAB/Simulink, where hardware layer that consists of two senders termed as publishers (in the yellow frame) and five receivers known as subscribers (in the pink frame). After the hardware layer is defined, it is very crucial to define the second and third layer i.e. software and communication layers. To do that the QoS is used by the system to exchange the data. Because the DDS is a content-aware middleware, it permits appending different QoS policies for each data type and treats each type in a unique way instead of applying the same policy to the whole data stream. For this purpose, a new library of QoS profile is created to control the data exchange. This feature helps to achieve the TSN capabilities for the network. The QoS policy defines a distinct set of rules that control how the data will be sent and handled by the infrastructure. To attain the TSN features in the network, multiple profiles have been designed, and a library is built. This library is used as the central QoS for the complete model. To define the profile's zone of operation a fourth layer, i.e. the function layer, is represented by the domain participant in the network (shown in a blue frame in Fig. 3). This layer prescribes where the TSN model is enabled.

C. QoS PROFILE DESIGN FOR TSN

The QoS profiles in the library are explained as follows. The time synchronization feature of TSN is fulfilled when all publishers or subscribers belong to the same domain and being the domain participants share the same time reference. For traffic scheduling, prioritization and system configuration (path reservation, bandwidth, time slots), scheduling policies

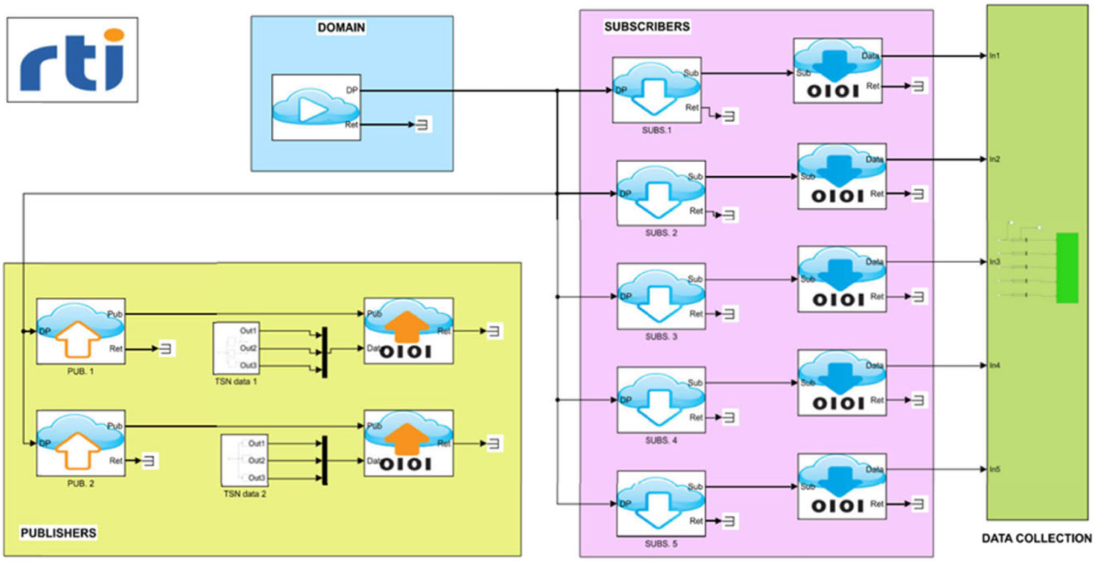


FIGURE 3. DDS Simulink model for general data structure with two publishers and five subscribers for TSN implementation.

such as Round Robin (RR), Earliest Deadline First (EDF) and Highest Priority First (HPF) are utilized. The EDF is used for the proposed model because it allows for optimizing the exploitation of the processing capabilities over the fixed priority scheduling policy and QoS library. In this policy, the priority of specific data can be decided dynamically based on the latency budget and deadline. Hence, it is ensured that the data packet is neither lost nor delayed beyond allowable (design) limits.

There are two separate profiles for defining latency and throughput in the library which help in prioritizing the latency-sensitive data. It specifies the time period within which DDS must distribute the information. This time period starts from the moment data is written by publisher until it is available in the subscriber's data cache ready for use by the readers. The throughput profile also helps in defining maximum throughput and preventing peak bursts. These QoS profile provides control over the temporal properties of data, while ensuring that bandwidth is exploited optimally.

A library is created by utilizing the listed QoS profiles. However, the new library is based on the built-in QoS profiles that are used to configure Domain Participants, Data Writers, Data Readers and Topics. For defining the latency and throughput Generic.StrictReliable profile is used. This profile basically guarantees delivery of every published sample. It ensures data will not be overwritten, regardless of HistoryQoSPolicy's depth. It confirms in-order delivery of every published sample and retransmission of every lost sample. Deterministic TSN frames are sent over layer 2 Ethernet. This means that routing is done using MAC address and not IP.

When a TSN schedule is created the Destination MAC address is replaced by a unique multicast MAC address which serves as an identifier for complete network. The sender uses this address as the Destination address. This also allows for a schedule to configure switches to deliver a single transmitted frame to multiple receivers.

TSN also typically ensures the data is sent on time and in order. The TSN toolkit API therefore adds 22 bytes information as a header to the payload of a TSN frame as shown in Table 1.

Critical values include 'Sequence Number' and 'Transmission Timestamp'. The Sequence Number increments by one on each transmission. This allows the receiver to identify if the frame is duplicated or lost. The Transmission Timestamp verifies that the transmission occurred during the time window provided by the schedule. In fact, the receiver can even subtract the transmission timestamp from the time of reception to calculate the transmission time of the frame through the network.

Low latency and high throughput are the extensions of this QoS profile. To achieve a fine-grained tuned performance certain parameter like packet size, maximum burst size, resolution and latency budget duration are added and override.

D. IMPLEMENTATION OF THE TSN MODEL USING DDS

To implement the TSN model, the Application Programming Interface (API) for the DDS middleware is used. It provides the necessary tools to integrate with different simulation and analysis software with support of several programming languages such as C, C++, and JAVA.

The DDS Simulink blocks and MATLAB classes use RTI Connext DDS. Blocks are added to a Simulink model that let the model interact with other DDS participants. During the simulation, The MATLAB/Simulink Coder generates C/C++ code from the model, and the generated code from the DDS blocks conform to the RTI Connext DDS API. This code is then compiled and executed on the platform supported by RTI Connext DDS or RTI Connext Micro DDS.

TABLE 1. **Payload of TSN frame.**

Chronos Stream	Chronos Header	UNSIGNED8 [3] UNSIGNED:8 UNSIGNED:8 UNSIGNED:8	Reserved if using Chronos Ether Type or if using OUI Ether type 0X00 0X80 0X2F
		UNSIGNED:4	Subtype
		UNSIGNED:4	Version
		UNSIGNED:32	Reserved
		UNSIGNED:1	More frames for this stream?
		UNSIGNED:4	Reserved
		UNSIGNED:11	Size of stream data in this frame (excluding Timestamp and CRC)
		UNSIGNED:32	Sequence Number
	Time stamp	UNSIGNED:64	Transmission time stamp
	Stream Data	UNSIGNED:8[22….1478]	Chronos Stream data

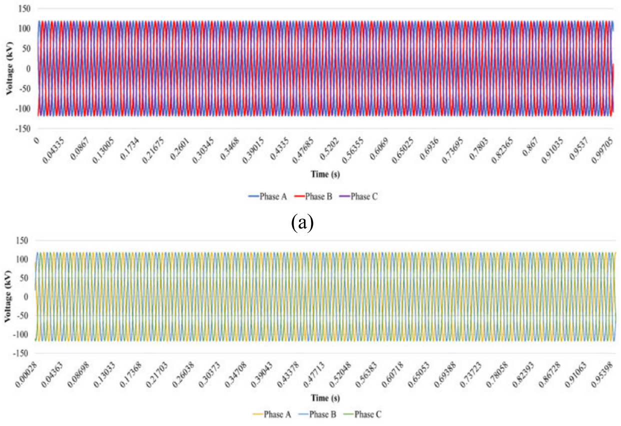
TABLE 2. **OPOS components and priorities in different levels.**

Publishers	Subscribers	Data
Pub. 1	Subs. 1	W13, W12, W11

All possible scenarios unicast, multicast and levels of prioritization amongst publishers (level I), subscribers (level II) and data (level III) are discussed in the next section.

IV. COMMUNICATION NETWORK DESIGN AND SIMULATION RESULTS FOR A WIND FARM MONITORING APPLICATION CASE

In the last section TSN model is established, its characteristics and configuration are presented. The communication network infrastructure is deployed in a testbed environment consisting of MATLAB version R2018a, Windows 10 Pro 64bit operating system, with Intel(R) Xeon(R) CPU E3-1241 v3 3.5GHz and 32.0 GB RAM and RTI Connext DDS version 5.3.1 with RTI DDS Library x64Win64VS2017 and MINGW64 C++ compiler. To validate the protocol in a real-time, the TSN model is provided with both three-phase point-of-wave data and synchrophasor data of a wind farm in three operational states: steady, transient and oscillatory. The data is replayed in real-time to demonstrate the features of the proposed simulation prototype. The windfarm data comes from stored records of a Disturbance Fault Recorder (DFR) and a PMU and corresponds to a site where sub-synchronous control interactions (SSCI) which are now referred to as forced oscillations [25]. This case is of interest because the harmonic content of SSCIs have shown to be beyond the Nyquist limit provided by most available PMUs (e.g. 15 Hz for a 30 Hz reporting rate). Using data with lower rate might result in corrupt estimates from monitoring applications [26]. Hence, detection of SSCI beyond this limit would require transfer of higher sampled data, such as the 5760 Hz data available from the DFR, however, this is not currently the industry practice. The simulation results aim to show that the application of TSN model helps easing the challenges of transmitting this data, while the challenge of SSCI detection from it is out of the scope of this paper – for current approaches see [25] and for a study on SSCI detection from the data used below [28].



(a)

(b)

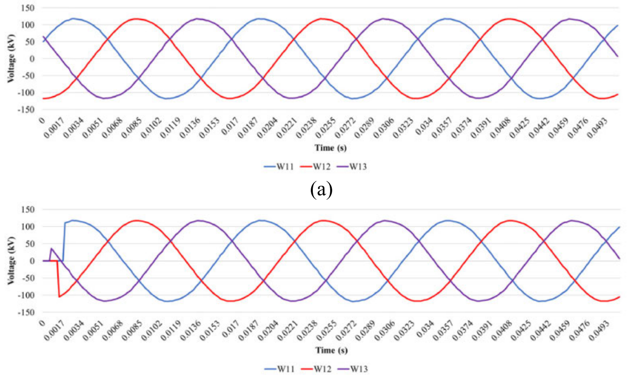
FIGURE 4. **Data waveforms being transferred from: a) Publisher 1; b) Publisher 2.**

The per-unit three-phase voltage waveforms at the main AC bus of a wind farm are used replay data in the communication network design simulations. Figure 4 represents the data waveforms which are sent by Publisher 1 and Publisher 2. They are denoted by $W_{1i}\,(i = 1, 2, 3)$ and $W_{2i}\,(i = 1, 2, 3)$, respectively.

A. CASE I: ONLY PUBLISHER ONLY SUBSCRIBER (OPOS)

This is the most basic case scenario of any network, where there is only one publisher (sender) and one subscriber(receiver) in the entire domain. In this scenario, there is no need of TSN protocol for levels I (publishers) and level II (subscribers) prioritization. But in case there are multiple data available, then this requires that TSN protocol to be implemented for level III (data) prioritization. This is needed where certain parts of data or a certain type of data has more importance or is more time critical than the others. Table 2 represents OPOS scenario where there is only one publisher i.e. Pub.1 and only one subscriber i.e. Subs.1 and data packet consists of three voltage waveforms (W11, W12, and W13) prioritized as shown.

Figure 5 illustrates the waveforms at publisher and subscriber ends, respectively. Fig. 5(a) shows the voltage waveforms sent by Pub.1 (W11, W12, W13) all at the same time instance. As shown in Fig. 5(b), Subs.1 subscribes to Pub.1 and follows the developed QoS profile. The voltage waveforms are prioritized as shown in Table 1 and each waveform is separated by a time frame.



(a)

(b)

FIGURE 5. **Only Publisher Only Subscriber (OPOS) scenario; a) Pub. 1, b) Subs. 1.**

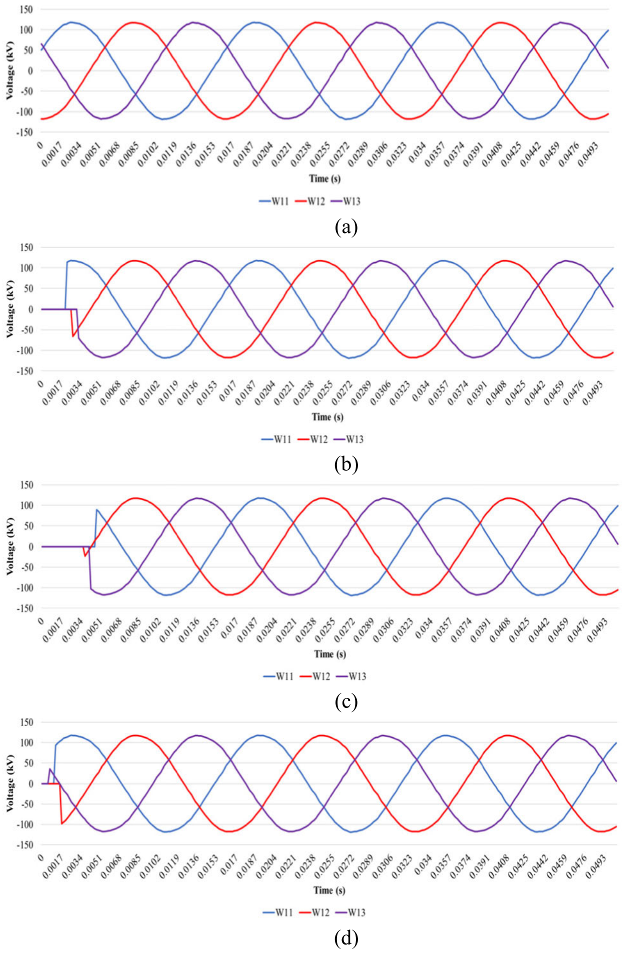
TABLE 3. **OPMS components and priorities in different levels.**

Publishers	Subscribers	Data
Pub. 1	Subs. 3	W13, W11, W12
	Subs. 1	W11, W12, W13
	Subs. 2	W12, W13, W11

B. CASE II: ONLY PUBLISHER MANIFOLD SUBSCRIBER (OPMS)

In this scenario, there is only one publisher which has information or data to share but the number of subscribers is multiple. It introduces the need of TSN protocol at levels II (subscribers) and level III (data) in case of multiple data packets. Since two levels of TSN are implemented, it is more complex and hierarchical than the previous case scenario i.e. OPOS. Table 3 corresponds to OPMS where there is only one publisher i.e. Pub.1 and three subscribers i.e. Subs.1, Subs.2 and Subs.3 which are prioritized as shown. The priority within data packet is also listed for each subscriber.

Figure 6 represents the waveforms at Pub.1 and Subs.1, Subs.2 and Subs.3 which are participating in this scenario. Fig.6(a) shows the three -phase voltage waveforms sent by Pub.1. In Fig. 6(b), 6(c), and 6(d) the waveforms received by subscribers Subs.1, Subs.2 and Subs.3 respectively are illustrated. The subscribers in the network are prioritized as Subs.3> Subs.1 > Subs.2 as per their time frames listed in Table 3. For Fig. 6(b) i.e. Subs.1 needs the Pub.1 waveform data in the order W13> W11> W12 but its own priority as the receiver is number 2 in the established network and it can be seen as a delay. Similarly, For Fig. 6(c) i.e. Subs.2 subscribes Pub.1 waveform data in the order W12>W13> W11 and it has the least priority amongst the receivers. While for Fig. 6(c) i.e. Subs.3 requires the Pub.1 waveform data in the order W13>W11>W12 asserting highest priority in the network. Each subscriber is independent of other in terms of received data and can be freely prioritized using the developed QoS library.



(a)

(b)

(c)

(d)

FIGURE 6. **Only Publisher Manifold Subscriber (OPMS) scenario; a) Pub. 1, b) Subs. 1, c) Subs. 2, d) Subs. 3.**

TABLE 4. **MPOS components and priorities in different levels.**

Publishers	Subscribers	Data
Pub. 2	Subs. 4	W22, W21, W23
Pub. 1		W13, W12, W11

C. CASE III: MANIFOLD PUBLISHER ONLY SUBSCRIBER (MPOS)

In this case scenario, there are manifold publishers to share information or from which data needs to be collected but only one subscriber. This limits the protocol as a subscriber can select only one publisher at a time i.e. it cannot read the data of every publisher in the domain simultaneously though it can prioritize the data freely for every publisher separately. Hence only level III TSN protocol is implemented in this case scenario. Table 4 represents the components that participate in this case scenario and their priorities.

Figures7(a), 7(b), and 7(c) show the waveforms at Pub.1, Pub.2, and Subs.4 ends, respectively. In Fig.7(a) & 7(b) two different voltage waveforms are shown which are read by

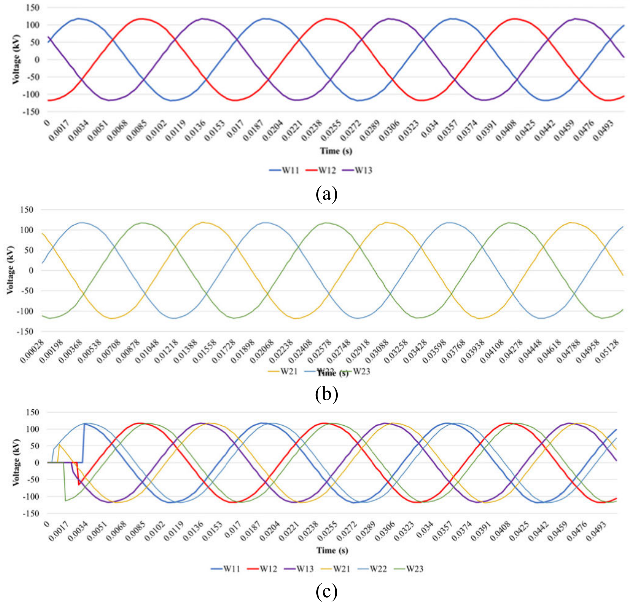


FIGURE 7. Manifold Publisher Only Subscriber (MPOS) scenario; a) Pub.1, b) Pub. 2, c) Subs. 4.

TABLE 5. MPMS components and priorities in different levels.

Publishers	Subscribers	Data
Pub. 1	Subs. 3	W11, W13, W12
	Subs. 1	W11, W13, W12
	Subs. 2	W11, W13, W12
Pub. 2	Subs. 5	W21, W23, W22
	Subs. 4	W21, W23, W22

Subs.4 as shown in Fig7(c). Subs.4 reads data from Pub.2 & then Pub.1 respectively and prioritizes data waveforms as listed in Table 4 within given time frames. In Fig. 7(c), Subs.4 subscribes the three-phase voltage for Pub.2 and then Pub.1. For Pub.2 it receives voltage data for phase 2 then one followed by three and presented as W22>W21> W23. Similarly, for Pub.1 the order of obtaining the voltage data is W13>W12>W11. This is clearly visible as delays between data waveforms in order of the assigned priorities.

D. CASE IV: MANIFOLD PUBLISHER MANIFOLD SUBSCRIBER (MPMS)

This case scenario presents the complete implementation of the TSN protocols at all the three levels i.e. at publisher level (level I), subscriber level (level II) and the data level (level III). As the name suggests it has manifold publishers and manifold subscribers which may or may not have multiple data packets. Table 5 lists all the publishers and subscribers in order of their priorities, also it is important to note that subscribers are not bound to receive data from every publisher in the network rather its optional and flexible. Pub.1 is subscribed by Subs.1, Subs.2, and Subs.3 whereas Pub.2 is subscribed by Subs.4 and Subs.5

Figure 8 illustrates the waveforms at publishers and subscribers ends for this case scenario. As shown in this Fig.8(a) and 8(b), Pub.1 and Pub.2 send out the two different voltage waveforms at the same time, respectively. From Fig. 8(c) till 8(g) each subscriber in the network and its related data is presented. In Fig. 8(c) Sub. 1 is shown and it chooses

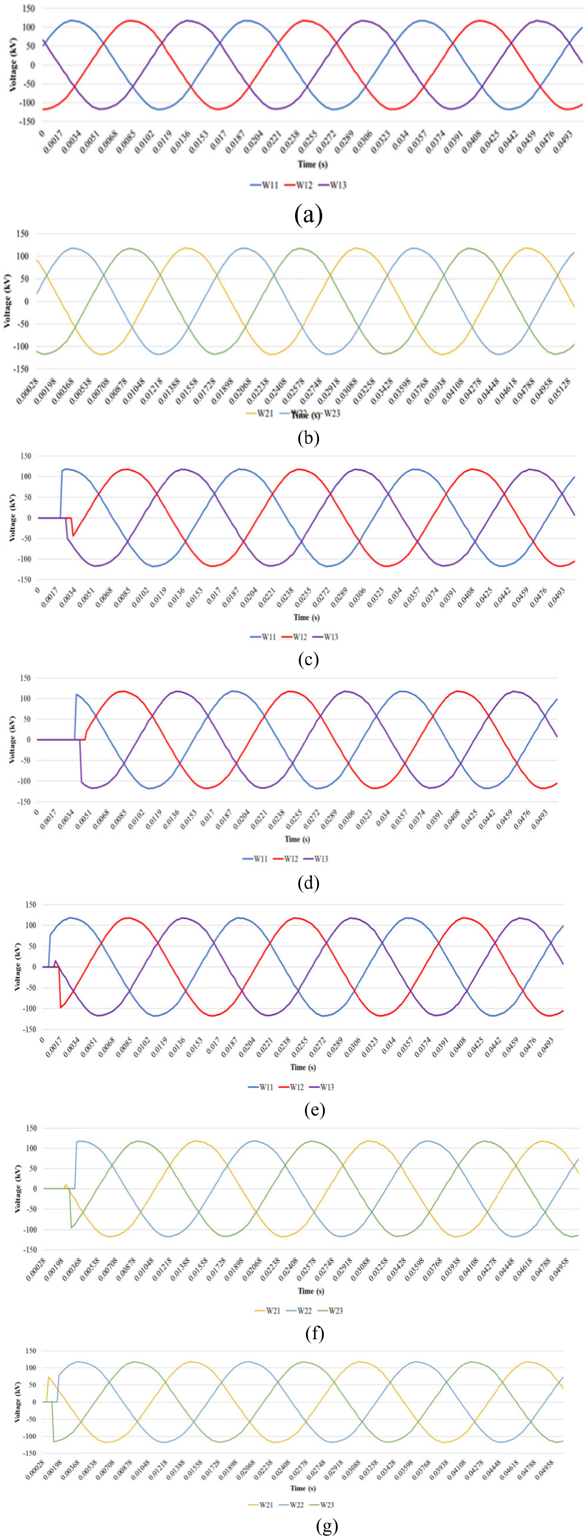


FIGURE 8. Manifold Publisher Manifold Subscriber (MPMS) scenario; a) Pub.1, b) Pub. 2, c) Subs. 1, d) Subs. 2, e) Subs. 3, f) Subs. 4, g) Subs. 5.

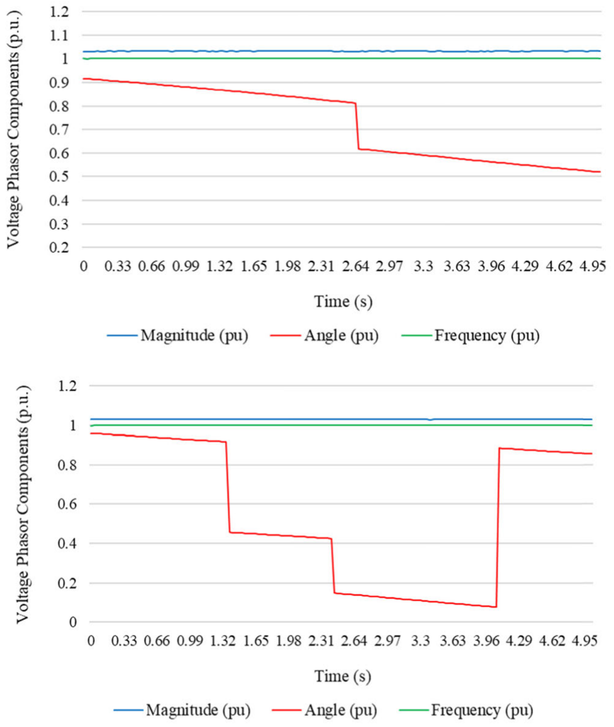


FIGURE 9. **The same data in PMU mode.**

to subscribe data of Pub.1. However, Sub. 1 is assigned the second priority by Pub.1 and the data is prioritized in the order W11>W13> W12. In Fig. 8(d) Sub. 2 is presented and it also subscribes Pub.1 data. Pub.1 assigns Sub. 2 with least priority amongst three subscribers. Its related data is ordered as W11>W13> W12. For the last subscriber of Pub.1 i.e. Sub. 3 has the highest priority, the data is arranged as W11>W13>W12.

The waveforms received by Subs.4 and Subs.5 which are subscribed to Pub.2, are shown in Fig.8(f) and 8(g). In Fig.8(f), Subs.4 is shown with data in order W21>W23> W22. and Fig.8(f) has data prioritized as W21>W23>W22 respectively, but Subs.5 receives data prior to Subs.4 by Pub.2.

This case scenario represents a near-to-real-life use case where there are many information nodes that unicast or multicast their information to fellow nodes in the network, and therefore, network traffic and its shaping is required.

E. PMU DATA TRANSPORT

The TSN Model has been validated by means of four different case scenarios as discussed above. Furthermore, the data used for TSN model can also be used to transport Phasor Measurement unit (PMU) data as shown in Fig.9 for the same nodes, this data comes from PMUs installed in the same locations of the DFRs.

This example shows that the TSN mode can handle both three phase measurements and PMU data. Because this PMU only provides positive sequence phasors and frequency, more data can be made available through the TSN-based network, and thus, the proposed method can be used for both three-phase and PMU data. In addition, because TSN provides an alternative to GPS technology, this would allow to bring redundancy to PMU data timestamps by using both GPS-based and TSN-based timing sources on it. Hence, if TSN integrated with PMUs, this can help to increase security to spoofing attacks [27] by implementing correction algorithms on the timestamp. This would make the data much more secure and can be used in applications such as smart military or civilian microgrids.

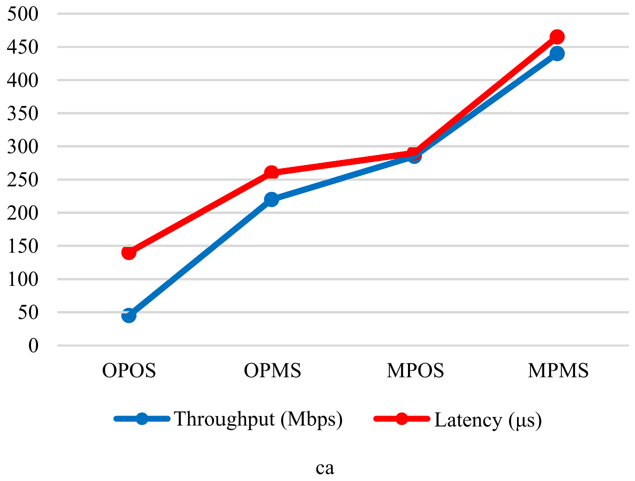


FIGURE 10. **Latency and throughput for all four case scenarios.**

F. PERFORMANCE

The performance of the communication network and DDS middleware are benchmarked by calculating the corresponding latency and throughput for each case scenarios discussed before. It is observed that the proposed framework maintains satisfactory latency, even at high message rates. Figure 10 represents that latency and throughput at the sampling rate of 100K samples per second for all case scenarios. The latency increases with throughput. But it remains under 500 microseconds, which is well beyond even the most common PMU reporting time of 33.33 milliseconds for each PMU packet. This means that even though data is being transmitted at ultra-fast sample rates, the proposed approach delivers three phase measurements at a faster rate than a PMU can even report it. It can be concluded that OPOS is the best-case scenario with latency of 140 microseconds while MPMS is the least one in terms of latency of 465 microseconds since the network has all levels of prioritization involved.

V. CONCLUSION

In this paper, the design and implementation of a multi-level Time Sensitive Networking (TSN) protocol based on a real-time communication platform utilizing Data Distribution Service (DDS) middleware for data transfer of synchronized three-phase power system measurements is presented. The developed protocol provides the capability to shape the network traffic and prioritize the time-critical data. The utilized publisher-subscriber scheme provides reliable and flexible communication while eliminating the network congestion and shaping the traffic. The QoS library design and implemented for this work includes profiles such as high throughput, low latency, reliable streaming, alarm event, and last value cache, and was developed to provide integration of TSN capabilities. The performance of the developed protocol was tested and validated by using real-time data, i.e. replayed the voltages and currents of a wind farm within different case scenarios. It was also shown how PMU data can be transported using the same approach, reflecting that the proposed TSN-enabled protocol can handle ultra-fast sampled data in a secure manner without any impact on latency, while at the same time providing a viable alternative to GPS technology for time-synchronization. Satisfactory results are obtained for latency and throughput parameters of TSN at high message rates at the sampling rate of 100K samples per second. A security layer can be added to this protocol as a future work to make it more secure and cryptic and be able to use it in more information-sensitive environments.

REFERENCES

[1] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, Feb. 2016.

[2] M. A. Uusitalo, "Global vision for the future wireless world from the WWRF," *IEEE Veh. Technol. Mag.*, vol. 1, no. 2, pp. 4–8, 2006.

[3] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, "Future Internet research and experimentation: The FIRE initiative," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 3, pp. 89–92, Jul. 2007.

[4] K. Paridel, E. Bainomugisha, Y. Vanrompay, Y. Berbers, and W. D. Meuter, "Middleware for the Internet of Things, design goals and challenges," *Electron. Commun. EASST*, vol. 28, pp. 1–6, Jun. 2010.

[5] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT urban scenarios," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3558–3567, Oct. 2013.

[6] *Contiki*. Accessed: Jun. 15, 2018. [Online]. Available: http://www.contiki-os.org/

[7] *Mbed*. Accessed: Jun. 18, 2018. [Online]. Available: http://mbed.org/technology/os/

[8] *RIOT*. Accessed: Jun. 22, 2018. [Online]. Available: http://www.riot-os.org/

[9] *FreeRTOS*. Accessed: Jul. 12, 2018. [Online]. Available: http://www.freertos.org/

[10] *Embedded Linux*. Accessed: Jun. 25, 2018. [Online]. Available: http://elinux.org

[11] *OpenWSN*. Accessed: Aug. 1, 2018. [Online]. Available: http://openwsn.atlassian.net

[12] *TinyOS*. Accessed: Aug. 5, 2018. [Online]. Available: http://www.tinyos.net/

[13] A. Katasonov, O. Kaikova, O. Khriyenko, S. Nikitin, and V. Y. Terziyan, "Smart semantic middleware for the Internet of Things," in *Proc. INSTICC*, 2008, pp. 169–178.

[14] Y. Ni, U. Kremer, A. Stere, and L. Iftode, "Programming ad-hoc networks of mobile and resource-constrained devices," in *Proc. SIGPLAN PLDI*, Jun. 2005, vol. 40, no. 6, pp. 249–260.

[15] H. Zhou, *The Internet of Things in the Cloud: A Middleware Perspective*, 1st ed. Boca Raton, FL, USA: CRC Press, 2012.

[16] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2013.

[17] R. Meier and V. Cahill, "STEAM: Event-based middleware for wireless ad hoc network," in *Proc. 22nd Int. Conf. Distrib. Comput. Syst. Workshops*, 2002, pp. 639–644.

[18] *RTI Whitepaper, Data Centric Middleware*. Accessed: Aug. 15, 2018. [Online]. Available: http://www.rti.com/docs/RTI_Data_Centric

[19] D. C. Schmidt and H. van't Hag, "Addressing the challenges of mission-critical information management in next-generation net-centric pub/sub systems with OpenSplice DDS," in *Proc. IEEE Int. Symp. Parallel Distrib. Process.*, Miami, FL, USA, Apr. 2008, pp. 1–8.

[20] M. Pahlevan and R. Obermaisser, "Evaluation of time-triggered traffic in time-sensitive networks using the OPNET simulation framework," in *Proc. 26th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, Cambridge, U.K., 2018, pp. 283–287.

[21] P. Pop, M. L. Raagaard, S. S. Craciunas, and W. Steiner, "Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 1, no. 1, pp. 86–94, Dec. 2016.

[22] *Time-Sensitive Networking Task Group*, IEEE Inc., Piscataway, NJ, USA, 2016.

[23] *IEEE 802.1 Audio/Video Bridging*, IEEE Standard 802.1, AVB Task Group, 2017.

[24] H. Hooshyar and L. Vanfretti, "A SGAM-based architecture for synchrophasor applications facilitating TSO/DSO interactions," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Washington, DC, USA, Apr. 2017, pp. 1–5.

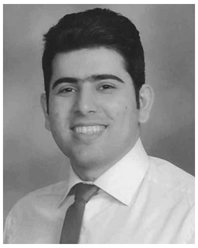
[25] North American Electric Reliability Corporation. (Sep. 2017). *Reliability Guideline—Forced Oscillation Monitoring and Mitigation*. [Online]. Available: https://www.nerc.com/comm/PC_Reliability_Guidelines_DL/Reliability_Guideline_-_Forced_Oscillations_-_2017-07-31_-_FINAL.pdfL

[26] L. Vanfretti, S. Bengtsson, V. S. Perić, and J. O. Gjerde, "Effects of forced oscillations in power system damping estimation," in *Proc. IEEE Int. Workshop Appl. Meas. Power Syst. (AMPS)*, Aachen, Germany, Sep. 2012, pp. 1–6.

[27] M. S. Almas, L. Vanfretti, R. S. Singh, and G. Margret Jonsdottir, "Vulnerability of synchrophasor-based WAMPAC applications' to time synchronization spoofing," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Portland, OR, USA, Aug. 2018, p. 1.

[28] L. Vanfretti, M. Baudette, J.-L. Domínguez-García, M. S. Almas, A. White, and J. O. Gjerde, "A phasor measurement unit based fast real-time oscillation detection application for monitoring wind-farm-to-grid sub-synchronous dynamics," *Electr. Power Compon. Syst.*, vol. 44, no. 2, pp. 123–134, 2016. doi: 10.1080/15325008.2015.1101727.

TANUSHREE AGARWAL received the B.Tech. degree in electronics and instrumentation engineering from Kurukshetra University, India, in 2008, the M.E. degree in electronics and product designing technology (EPDT) from the Punjab Engineering College, India, in 2010, and the Ph.D. degree from the Phillip M. Drayer Department of Electrical Engineering, Lamar University, Beaumont, TX, USA, in Spring 2019, where she has been actively involved in the projects with the Lamar Renewable Energy Microgrid (LAREM) Laboratory, since 2016. Her research interests include cyber-physical systems, synchrophasor and their related networks and communication-embedded systems, and control systems.



PAYAM NIKNEJAD (S'09) received the B.S. degree in electrical engineering from Shahid Beheshti University, Tehran, Iran, in 2011, the M.S. degree in power engineering from the University of Guilan, Rasht, Iran, in 2013, and the Ph.D. degree in electrical engineering with a thesis in resilience analysis and enhancement of power microgrids from the Phillip M. Drayer Department of Electrical Engineering, Lamar University, Beaumont, TX, USA, in 2019, where he has been an Active Member of the Lamar Renewable Energy Microgrid (LAREM) Laboratory, since 2016. His research interests include power grid resilience, renewable energies, microgrids, and power electronic converters.

M. R. BARZEGARAN received the Ph.D. degree in electrical engineering from Florida International University, in 2014.

He was a Postdoctoral Fellow with Florida International University, in summer 2014. He is currently the Director of the Lamar Renewable Energy Microgrid Laboratory with several bachelor's, master's, and Ph.D. research assistants on building resilient microgrid. He is the author of more than 70 journals and conferences, one book chapter, and several technical reports in the general area of electric machine drives and magnetics. His research interests include design, control and condition monitoring of electrical machines, as well as alternator and power system components, hardware in the loop control of electric machines, power electronics converters, embedded systems programming for power system, electric and hybrid electric vehicles, wireless power transfer, electromagnetic compatibility in power components, life assessment of electrical power components, and computer-aid simulation of power components. He was a recipient of fellowship 2014 and the Fellowship Award from the college of engineering. He was also a recipient of the Outstanding Graduate Award from Florida International University, class of 2014. He served as a Panelist for the National Science Foundation (NSF), DOE, the Associate Chair, and the Chair of the several IEEE Transactions and conferences and a Reviewer of numerous journals in the IEEE, Elsevier, Wiley, and so on.

LUIGI VANFRETTI (S'14) received the M.Sc. and Ph.D. degrees in electric power engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2007 and 2009, respectively. He was with the KTH Royal Institute of Technology, Stockholm, Sweden, as an Assistant, from 2010 to 2013, and an Associate Professor (Tenured), and a Docent, from August 2013 to August 2017, where he leads the SmarTS Lab and research group. He was also with Statnett SF, the Norwegian Electric Power Transmission System Operator, as a Consultant, from 2011 to 2012, and a Special Advisor in R&D, from 2013 to 2016. In August 2017, he joined the Rensselaer Polytechnic Institute, where he is currently an Associate Professor with the Department of Electrical, Computer, and Systems Engineering, and where he has established the ALSET Lab, a real-time hardware-in-the-loop simulation facility for cyber-physical system experimentation and research. His research interests include cyber-physical systems modeling, simulation, stability and control, and synchrophasor applications in power systems.

• • •