



# Article Design and Application of a Distribution Network Phasor Data Concentrator

Weiqing Tao <sup>1</sup>, Mengyu Ma <sup>1,\*</sup>, Chen Fang <sup>2</sup>, Wei Xie <sup>2</sup>, Ming Ding <sup>1</sup>, Dachao Xu <sup>3</sup> and Yangqing Shi <sup>3</sup>

- <sup>1</sup> School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, China; wqtao@hfut.edu.cn (W.T.); mingding56@126.com (M.D.)
- <sup>2</sup> Electric Power Research Institute, State Grid Shanghai Municipal Electric Power Company, Shanghai 200437, China; fangc02@gmail.com (C.F.); samedson@hotmail.com (W.X.)
- <sup>3</sup> CSG Smart Grid Electrical Technology Co., Ltd., Hefei 230080, China; rockysroll@126.com (D.X.); shiyangqing@csg.com.cn (Y.S.)
- \* Correspondence: mamengyu2013@163.com

Received: 8 January 2020; Accepted: 20 April 2020; Published: 24 April 2020



Abstract: The wide area measurement system (WAMS) based on synchronous phasor measurement technology has been widely used in power transmission grids to achieve dynamic monitoring and control of the power grid. At present, to better realize real-time situational awareness and control of the distribution network, synchronous phasor measurement technology has been gradually applied to the distribution network, such as the application of micro multifunctional phasor measurement units ( $\mu$ MPMUs). The distribution network phasor data concentrator (DPDC), as a connection node between the µMPMUs and the main station, is also gaining more attraction. This paper first analyzes the communication network structure of DPDCs and µMPMUs and compares and analyzes the differences in the installation locations, functions, communication access methods and communication protocols of the phasor technology devices of the distribution network and the transmission network. It is pointed out that DPDCs not only need the functions of data collection, storage, and forwarding like transmission network PDCs, but also should be able to access more  $\mu$ MPMUs, and can aggregate the phasor data of the same time scale from  $\mu$ MPMUs by different communication methods. The communication protocol selected by DPDC should be expanded to support remote control, telemetry, fault diagnosis and other functions of distribution automation. The application requirements of DPDCs are clarified, and the key indicators of DPDCs are given as a method to evaluate the basic performance of DPDCs. Then, to address the problems of more µMPMU access, abnormal communication, and data collection with different delays that DPDC encountered, a DPDC that considers multiple communication methods is designed. Based on the Linux system and the libuv library, the DPDC is designed with event-driven mechanism and structured programming, runs multiple threads to implement multitasking, and invokes callbacks to perform asynchronous non-blocking operations. The DPDC test system and test methods are designed. The performance of the designed DPDC is evaluated through the test and the test results are analyzed. Lastly, its real-world application is disclosed, which further confirmed the value of our DPDC.

Keywords: distribution network; PDC; key performance indicators; test

## 1. Introduction

With the distributed energy such as photovoltaics and wind power, flexible load and electric vehicles connected to the distribution network on a large scale, the stable operation of the distribution network faces new challenges in the aspects of the difficulties to quickly and accurately carry out

the state estimation, fault diagnosis and active control using monitoring and controlling methods of the traditional distribution network. To cope with these challenges, the synchronous phasor measurement technology is introduced to the distribution network [1,2] with reference to the wide area measurement system (WAMS) widely used in transmission networks. The micro multifunctional phasor measurement unit ( $\mu$ MPMU) is a miniaturized phasor measurement unit (PMU) that combines the functions of power distribution monitoring terminals [3], which can also be called a distribution network phasor measurement unit (DPMU) or micro phase measurement unit ( $\mu$ PMU) [4].  $\mu$ MPMUs provide global positioning system (GPS)/BeiDou navigation satellite system (BDS)-based measurement data so that power signals at different locations can be put together for comparison and analysis, such as phase angle.  $\mu$ MPMUs are mainly used to solve the problems of sensing and controlling in the distribution network, including fault information extraction and fault diagnosis, state estimation, voltage control, reactive power optimization and active control [4,5].

In addition to PMUs, the transmission network WAMS also includes components such as phasor data concentrators (PDCs) and communication networks [6,7]. According to the depiction of IEEE C37.244-2013 [8], the primary and most important role of the PDC is to aggregate and forward data of multiple PMUs according to timestamps. Based on this guidance, IEEE introduced the PDC standard IEEE Std C37.247-2019 [9] in 2019, which specifies the requirements for PDC. With the development of technology in this area, PDC further has the ability to process and store data and configure more functions according to different requirements. Although the current application of  $\mu$ MPMUs in the distribution network is increasing, the corresponding distribution network phasor data concentrator (DPDC) has not received sufficient attention.

In terms of PDC design, the PDC designed in [10] has a variety of data push logics and has a good performance in ensuring data integrity and latency but there is no targeted design for the PDC for the application scenario of the distribution network. Reference [11] proposed a software PDC that used a reliable communication protocol to establish a database system for efficient data storage and access. It is used as a low-cost alternative to large PDCs, while its application scenario is still in the transmission grid. Reference [12] discussed the design of PDC and the handling of anomalies. The anomalies handling is very meaningful, because DPDC may encounter these problems more frequently in the distribution network, but the article does not elaborate on the details of the anomalies handling scheme. Reference [13] proposed a software PDC for monitoring DER in a smart microgrid, which uses an adaptive compensation scheme to achieve an effective estimation of missing data elements, and proposed a monitoring unit to detect DER. Reference [14] considered the situation that more and more PDCs and PMUs were being installed in the distribution network, and proposed an active PDC with advanced functions to manage the delay according to the requirements of possible target applications. Reference [15] proposed a low-cost active PDC suitable for power distribution networks. The prototype of this PDC is based on the low-cost hardware platform SBC Raspberry Pi. Controlling costs is very important for the promotion of PDCs in the distribution network. At the same time, in order to meet the potential for increased equipment access, resources of a PDC should not be too limited. Reference [16] introduced an FNET/GridEye example of WAMS deployed in the distribution network. According to reports, the frequency monitoring network FNET/GridEye was originally developed in 2003 and was the first WAMS designed for distribution networks. Its sensors have the advantages of low cost and easy installation. Since FNET/GridEye relies on public networks for communication, its latency and other communication problems are more serious than WAMS for transmission networks.

The existing PDC solution is mainly for connecting to the PMUs in the transmission network WAMS. Although there are some discussions on PDCs suitable for distribution networks, there is a lack of consideration of the working environment and communication environment of  $\mu$ MPMU and DPDC in distribution networks and the lack of analysis of DPDC application requirements. Various software and hardware PDCs have been introduced in the above solutions but they may face problems such as fewer application scenarios, single functions, higher costs, and limited equipment platform

resources. To better implement the promotion of  $\mu$ MPMU in the distribution network, discussions on DPDC design and testing. The main contributions of this paper are:

- Explain the µMPMU-DPDC networking structure, analyze the application requirements of DPDC, and give the key performance indicators of DPDC;
- A hardware DPDC design method is proposed. Both software design and hardware selection take into account the need to meet the key performance indicators of DPDC;
- Proposed a DPDC key performance indicators test method and verified the performance of the designed DPDC through testing.

The structure of this paper is as follows: Section 2 introduces the networking structure of  $\mu$ MPMU-DPDC, analyzes the application requirements of DPDC, and provides key performance indicators of DPDC; Section 3 describes the design of DPDC, including software structure and hardware selection, and discloses software implementation details; Section 4 introduces the DPDC test environment and test tools, evaluates the DPDC's key performance indicators; Section 5 describes the DPDC's field installation and operation; Section 6 summarizes the article.

#### 2. DPDC Application Requirements Analysis

### 2.1. µMPMU-DPDC Networking Structures

The PMUs in the transmission network WAMS are mainly installed in important substations and power plants. In actual engineering, the PDC is generally installed in the same screen cabinet as the PMU. Each device gets timing information from a unified synchronous clock source. PMUs can be directly connected to the PDC or connected to the PDC via a switch. If the PDC is not equipped, PMUs are directly connected to the switch in the plant. The WAMS main station connects to the power dispatch data network through the pre-communication subsystem and receives data from PDCs or PMUs. It can be seen that the working environment of the PDCs and the PMUs in the transmission network WAMS is not bad, the communication connection between the devices is relatively stable, and PMU's clock signal source is also reliable. Limited by the number of important components monitored in the plant, the number of PMUs connected to the PDC in the station will not be much.

In contrast to the PDCs and PMUs in the above-mentioned transmission network WAMS,  $\mu$ MPMUs are installed in the substation, as well as the feeder, ring network cabinet, DER and other points [3]. The  $\mu$ MPMU outside the substation obtains the time signal through the external GPS antenna of the device. In addition to fiber optic Ethernet and Ethernet passive optical network (EPON), the communication method between the  $\mu$ MPMUs and the DPDCs or the main station also uses communication methods with a higher delay such as power line communication (PLC) or a wireless private network. Considering the installation location and communication mode of the  $\mu$ MPMUs and DPDCs, the networking modes of the  $\mu$ MPMU-DPDC can be divided into three types, as shown in Figure 1.

Figure 1a shows a centralized networking structure in which all the  $\mu$ MPMUs and the DPDC are located in the station. This structure is consistent with the PMU-PDC networking structures in the transmission network. Figure 1b1,b2 indicates that the  $\mu$ MPMUs are dispersedly installed on the ring main unit and each section of the feeder. The DPDCs are installed inside the station, and they communicate with  $\mu$ MPMUs via wired communication such as EPON or PLC. Figure 1c indicates that the  $\mu$ MPMUs are dispersedly installed at feeder line and DER connection points. The  $\mu$ MPMU communicates with the upper device through a built-in wireless communication module or an external customer premise equipment (CPE). The DPDC or the main station is connected to the core network to receive data from the  $\mu$ MPMU. In this case, the DPDC can be placed in a substation, core network or inside of a main station. If only a few scattered  $\mu$ MPMUs use wireless communication, such  $\mu$ MPMUs can communicate directly with the main station without going through the DPDC. This figure does not mean that the  $\mu$ MPMU monitoring DER must use wireless communication. Under the conditions of proper installation cost,  $\mu$ MPMU should choose low-latency fiber-optic communication.



**Figure 1.** μMPMU-DPDC networking structures. (**a**) centralized in the station; (**b1,b2**) wired distributed; (**c**) wireless distributed.

#### 2.2. Application Requirements

In addition to the different networking structures compared with the PMU-PDC in the transmission network, the  $\mu$ MPMU connected to the DPDC is different from the traditional PMU in terms of function, quantity, and communication method [3]. To meet the application requirements in the active distribution network, the  $\mu$ MPMUs have the functions of power distribution terminals, and the corresponding DPDCs also need to have certain functions of the distribution station. The DPDC can be used for data collection, and the collected  $\mu$ MPMUs data is sent to the main station, and the control commands from the main station are forwarded to the  $\mu$ MPMU. The DPDC can also directly have the monitoring function. By analyzing the data of  $\mu$ MPMUs, the DPDC controls  $\mu$ MPMUs to complete fault location, isolation, restore power to non-faulty areas, the DPDC then reports the processing results to the main station. Thereby the feeder automation is implemented.

To make the problem that DPDC needs to solve more clear, PMU and  $\mu$ MPMU, PDC and DPDC are compared from various aspects, and the comparison contents are shown in Table 1.

Through the comparison in the table, the following conclusions can be drawn regarding the application requirements of DPDC:

- (1) The number of µMPMUs is large and the number of points may increase as the number of important local nodes increases. A DPDC as the substation/node PDC should be able to connect more than 20 µMPMUs [17], and the dynamic data files can be stored for 14 days, and the transient data files more than 1000 pieces. Commercial PDCs can connect hundreds of PMUs, but their prices are thousands of dollars or more, and there is no price advantage in promoting them in the distribution network;
- (2) The installation locations of μMPMUs are scattered and the working environment is diverse. Compared with the traditional PMU, the μMPMU may face the problem of communication interruption and clock loss [11]. The DPDC needs to be able to correctly assemble the data and not discard the clock abnormal data, and also has the function of reconnecting the μMPMU;
- (3) The communication modes adopted by µMPMU are various, and the delay of different communication methods is different. Wireless communication may have an unstable connection

under different meteorological or spatial conditions, so in the case of using a synchronous time signal, the data frames of the same timestamp produced by different  $\mu$ MPMU may arrive at the DPDC at different times, after experiencing a significant delay. The DPDC should be able to tolerate a large arrival time difference of the same timestamp data between the  $\mu$ MPMUs, that is, to correctly collect data of different delays. Reference [18] summarized the delay of wireless communication, and reference [9] also introduced the delay of PMU data transmission using 4G LTE, at an average of 70 ms, and even 1 s in certain cases. This paper defines DPDC's ability to tolerate data time differences as time difference tolerance, with a reference value of 150 ms;

- (4) Considering the versatility of the μMPMU, the main tasks performed by different μMPMUs may be different, and different applications have different data delay requirements, ranging from 100 ms to 5 s [19]. The classification and priority transmission of data are helpful for increasing DPDC ability and improving communication efficiency. Traditional phasor transmission such as protocol IEEE C37.118.2-2011 [20] and GB/T 26865.2-2011 [21] cannot meet the requirements, and DPDC needs to support the extended protocol. IEEE C37.118.2-2011 is an internationally popular synchronous phasor transmission protocol. GB/T 26865.2-2011 is a synchronous phasor transmission protocol used by China's power system, and the contents are mostly the same. The table takes GB/T 26865.2-2011 as an example. The extended protocol is based on the GB/T 26865.2-2011 protocol, adding data priority, remote control commands, etc. The method is also applicable to IEEE C37.118.2-2011, more details can be seen in the paper [3];
- (5) A large number of μMPMUs accessed by DPDC will send phasor data at high frequency and will also send a large amount of file data and commands. To avoid affecting the real-time monitoring and active control capabilities of the entire system, DPDC should be able to efficiently process, aggregate and forward data. This requires the DPDC to have a lower processing delay [8], where the DPDC processing delay is defined as the DPDC data delay, indicating the arrival time of the last complete data message with a given timestamp and its leaving time difference from the DPDC, in the case where the data aggregation is normal. DPDC data delay should be within 3–10 ms [17];
- (6) Since the increases in the number of μMPMUs also leads to an increase in communication traffic, which further leads to an increase in storage footprint. A more directed way to reduce flow and storage space is via data compression. To avoid any data accuracy loss and data latency increase, it is beneficial to use a suitable compression method.

Device PMU		μΜΡΜU	PDC	DPDC
Locations	Substation; power plant	Substation; power distribution room; Substation transformer; feeder power pla line; DER		Substation; power distribution room
Main Functions	Synchronous phasor measurement	onous Synchronous phasor Aggregate and sor measurement; forward data ement distribution terminal		Aggregate and forward data; power distributing substation
Communication Modes	Optica fiber; Twisted pair	Optica fiber; twisted pair; wireless private network; power line carrier	Optical fiber; twisted pair	Optical fiber; twisted pair; wireless private network; power line carrier
Communication Protocols	GB/T 26865.2-2011; IEC/TR 61850-90-2-2016	GB/T 26865.2-2011; extended protocol	GB/T 26865.2-2011	GB/T 26865.2-2011; extended protocol

#### Table 1. Device comparison.

Based on the above analysis, it can be concluded that DPDC needs to have expanded functions and features in order to cope with various situations. It summarizes the main problems faced by DPDC and their corresponding extended functions and features, as shown in Table 2.

No.	Situation Faced	<b>DPDC</b> Extended Functions and Features
1	Accessing more $\mu$ MPMUs	DPDC should have strong access capabilities and keep costs low
2	µMPMUs adopt different communication methods	The DPDC should be able to aggregate data frames that have the same timestamp but arrive at the DPDC with a large time difference; when some devices are disconnected, the normally arriving data is not discarded
3	µMPMUs expand distribution terminal functions	In the communication protocol, power distribution automation functions such as remote control, telemetry, and fault diagnosis should be added.
4	Processing of large amounts of data of different kinds	DPDC should improve data processing capabilities and keep data delay low.

Table 2.	DPDC	extended	functions	and	features.
----------	------	----------	-----------	-----	-----------

This paper proposes the key performance indicators of DPDC, which are also the most basic requirements for DPDC, namely access capability, time difference tolerance, and data delay. The performance of DPDC consists of software performance and hardware performance. The hardware performance depends on the CPU, network port, running memory and hard disk. The software performance is affected by the program frame and operation mechanism. By testing the key performance indicators of DPDC, the performance of DPDC is judged [22].

#### 3. DPDC Design

#### 3.1. Software Architecture

In the software design of DPDC, to improve the high concurrency and asynchronous processing capability of the software, we adopt an event-driven mechanism and structured program design, run multiple threads to realize multi-tasking processing and use callbacks to complete asynchronous non-blocking operations. Figure 2 is a schematic diagram of the proposed DPDC considering multi-communication mode, which shows some major software functional modules inside the DPDC. The figure simplifies the  $\mu$ MPMUs, the communication network, and the main station. ① Indicates that the  $\mu$ MPMU communicates directly with the DPDC over a twisted pair. ② Indicates that the  $\mu$ MPMU communicates with the DPDC over the power line carrier. ③ Indicates that the  $\mu$ MPMU communicates with the DPDC over the wireless network. ④ Indicates that the  $\mu$ MPMU communicates with the DPDC over possibly other communication networks. The communication network from the DPDC to the main station is simplified to a dotted line. The green port on the DPDC boundary indicates the network port. The DPDC has more network ports.

First, the communication is initialized in the main function of the DPDC, including establishing a client to complete the connection with the  $\mu$ MPMU and establishing a server to listen for the connection from the main station. The open-source iPDC [23] establishes a thread for each connected PMU and the main station and has only one connection to each PMU or main station, and all types of data are transmitted over one connection. The difference is that DPDC only creates two child threads in addition to the main thread. In the main thread, DPDC establishes two TCP connections for each  $\mu$ MPMU and each main station, which serve as data pipes and command pipes. One sub-thread is responsible for establishing a TCP connection as a file pipe with each  $\mu$ MPMU, and the other is responsible for establishing a file pipe with the main station. This can avoid the impact of file transfer on real-time

data transmission, save the resources consumed by creating a large number of threads, reduce the time overhead of thread switching, and reduce the internal delay of DPDC.

We use libuv to establish a TCP connection because libuv is a multi-platform support library that focuses on asynchronous I/O. It uses the high-concurrency asynchronous model provided by the operating system. In Linux, epoll supports full-featured event loops [24]. Due to libuv's event-loop and callback mechanisms, callback functions can be invoked when the event occurs. When the connection is established, the uv\_read\_start function defined by libuv will automatically read the data from the stream and trigger the callback function. Since the DPDC and each µMPMU have dedicated pipes, the data received from a pipe is the corresponding  $\mu$ MPMU data. Inside the DPDC has shown in Figure 2, the blue lines indicate data pipes for transmitting data frames, the red lines indicate command pipes for transmitting command frames (CMD) and configuration frames (CFG), and the orange lines indicate file pipes for transmitting file command frames and file frames. The data pipes and the command pipes will remain connected after the communication is started. The file pipes will only be established when there is a file sending task and will be disconnected after the task is completed. In order to ensure the stability of the data pipes and command pipes between the DPDC and µMPMUs, the connection check module is used to check whether each connection is valid. Here we set a timer of 3 s to check the continuity of the connection. If it is determined that a connection has been disconnected, the connection is re-established. After the  $\mu$ MPMU restarts or the communication link is temporarily faulty, etc., the communication can be resumed through the connection check module.



Figure 2. DPDC architecture considering multiple communication methods.

When the data from the µMPMU enters the DPDC via the data pipe, the Read data module completes the reading of the data in response to the event. Similarly, the Read cmd module reads data from the command pipe, and the Read file module reads data from the file pipe. All kinds of data will be sent to the data check module first, and then checked and processed by the data check module and sent to the corresponding pipe data processing module. The data check module checks if the data has sticky or incomplete packets. Sticky packets mean that multiple small packets are placed in the same TCP packet. An incomplete packet means that a large packet is separated into multiple TCP packets. The worst case is that a TCP packet has a complete packet and an incomplete packet, and the remaining data is placed in the subsequent TCP packet for transmission. The data check module

decomposes and combines the data in which the above situation occurs and sends the processed data to each pipe data processing module according to the type. In general, the PMU and  $\mu$ MPMU disable the Nagle algorithm of TCP in order to ensure the uniform transmission of data frames in time, so that data frames do not have sticky packets and incomplete packets. The data check module checks the data frames with very little time and does not need to process them. Because configuration frames and file frames are longer than data frames, they are most likely to have sticky and incomplete packets, but their requirements for time are far less than the data frame, and the processing time is acceptable.

After the data check module, the data frame is sent to the data processing module of the  $\mu$ MPMU data pipe (D\_data\_pipe). The main function of this module is to aggregate the data frames of the  $\mu$ MPMU. First, it is judged whether the data frame is a compressed frame, and the decompressed data frame is sent to the data aggregation stage for processing. If no decompression is required, the data frame directly enters the data aggregation stage. The data aggregation stage produces DPDC data frames in the standard format, which are written to real-time (RT) data files (the file format refers to [21]) and sent to the data processing module of the main station data pipe (U\_data\_pipe). The data of  $\mu$ MPMU and DPDC are stored by using structure arrays. Figure 3 shows the cache structure of the DPDC.



Buffer end

Figure 3. The cache structure of the DPDC.

The structure array pdcData[] is used as the buffer of the DPDC, and the depth is max + 1. The value of max will affect the time difference tolerance of the DPDC. In theory, the time difference tolerance is proportional to the max. This value is not recommended to be set to a fixed value. It should be set according to the actual network environment where the DPDC is located. It is included in the configuration file sent by the local or the main station and is assigned when the device is initialized. The data structure members have integer haspmu, long integer soc, and fracsec, and array pmusData[]. Similarly, pmusData[] is a structure array with a length of n + 1. The value is the same as the number of  $\mu$ MPMUs. It is used to store the data frames of the  $\mu$ MPMU. The order of the array elements is the same as that of the  $\mu$ MPMU. The haspmu indicates the number of data frames of the  $\mu$ MPMU that this pdcData array element has stored. When has pnu = n + 1, this buffer is full and the data aggregation is completed. Proceeding to the next step, involved forming the collected data into a standard DPDC data frame, and then sending it to the master station and write the file. The soc and fracsec represent time, and their source can be local time or the  $\mu$ MPMU time-synchronized data frame. Which source to choose depends on whether there is a time-synchronized data frame in this buffer. If an element of pdcData[] has not received a time-synchronized data frame when it is about to be sent to the next stage, the DPDC's local time is assigned to it. When the cache is initialized, haspmu, soc, and fracsec are all cleared, and the status word stat of each element of pmusData[] is set to 0x8000, indicating that the data is not available and the rest of the data is cleared.

The program flow of the data aggregation stage is shown in Figure 4. The problem that will be solved during the data aggregation stage is also the abnormal communication situation that may

be faced: (i) when the communication connection is normal, some or all of the  $\mu$ MPMUs are out of synchronization and receive their data frames; (ii) some  $\mu$ MPMUs are disconnected from the DPDC and cannot receive their data frames; (iii) some  $\mu$ MPMUs are disconnected, and some  $\mu$ MPMUs with normal communication connections are out of synchronization. In the data aggregation stage, the thread lock mechanism is used to prevent data in the DPDC buffer from being overwritten by other threads.



Figure 4. Data aggregation program flow.

In Figure 4, the data frame of the i-th  $\mu$ MPMU ( $0 \le i \le n$ ) is first acquired, stored in pmusData[i], and then the local time is acquired. According to IEEE C37.118.2-2011 and GB/T 26865.2-2011, in the stat, bit 13 = 0 means data frame time synchronization, so it is judged according to stat whether the data frame time is synchronized. If the data frame time is synchronized, the process flow F1 will be entered, otherwise, the process flow F2 will be entered.

(1) The first step of F1 is to find the pdcData[] element with the same time as pmusData[i]. If the element is found, it is assumed to be pdcData[j] and proceed to flow F11. Put pmusData[i] into the corresponding pdcData[j].pmusData[i] and increase the value of pdcData[j].haspmu by 1. If the value of pdcData[j].haspmu is equal to n, it indicates that pdcData[j] has already aggregated the data frames of all µMPMUs. Process pdcData[j] means that the data contained in pdcData[j] is composed into a standard format DPDC data frame, and then pdcData[j] is cleared, and these data frames are all written to the file and sent to the U\_data\_pipe data processing module. Because 50

or 100 writes per second affect the life of the hard disk, the write operation does not have the same frequency as the framing. The program uses another large buffer to hold the data to be written and writes at regular intervals.

- (2) If pdcData[j] is not found, continue to look for elements with zero time. If an element that meets the criteria is found, the process flow F12 is performed, otherwise, the process flow F13 is performed. There may be two cases when the time is 0. The first is that the element has not yet obtained the data frame of the  $\mu$ MPMU, and the second is that the element has not yet obtained the time-synchronized data frame. In flow F12, assume that pdcData[k] is the element that meets the criteria and is closest to the cache header, store pmusData[i] into pdcData[k], and assign the synchronization time to it, and then process and judge pdcData[k].haspmu. When there is µMPMU disconnected communication, it will enter flow F13 because pmusData[i] with synchronization time cannot find the elements of the same time and elements without time in the whole buffer, indicating that all elements have data at this time. But they have not been sent yet, they are waiting for some  $\mu$ MPMUs data that has not been received. In order to control the delay of the data and avoid the overflow of the buffer, process the element with the earliest time, assuming it is pdcData[m], then pmusData[i] can be placed in the cache normally. Because pdcData[m] has not collected all the data, the empty pmusData[] element will keep the original value, except stat is 0x8000, the rest of the fields are all 0. According to the stat, the main station can know which µMPMU data is missing.
- (3) The first step in F2 is to look for the element that already has data but no i-th μMPMU data from the buffer header. If the eligible element is found, assume it is pdcData[p], ends the query, goes to flow F21, and puts pmusData[i] into pdcData[p]. If the pdcData[p] data has already aggregated all the data, it will be judged whether its time is 0 before processing. If it is 0, the local time obtained in the previous step is used. This is in order to deal with the extreme situation where all µMPMUs are out of sync.
- (4) If pdcData[p] is not found, continue to look for elements with time 0 and no i-th µMPMU data. If an element that meets the criteria is found, assume it is pdcData[q], go to flow F22, otherwise go to flow F23. Put pmusData[i] into pdcData[q]. If the pdcData[q] data has already aggregated all the data, assign the local time to pdcData[q] and process it. Similar to F13, F23 is also dealing with extreme situations. Some µMPMUs are disconnected, and some µMPMUs with normal communication connections are out of synchronization, the data will go to the F23 process. In order to control the delay of the data and avoid the overflow of the buffer, when the cache is full for the first time, the local time is assigned to the pdcData[m] at the top of the buffer, and then it is processed. Each time F23 is entered, the program will move down one element for processing.

The standard format DPDC data frame enters the U\_data\_pipe data processing module and is ready for transmission. Because the communication rate requirement of the main station to the DPDC may be different from that between the DPDC and the  $\mu$ MPMU, it is necessary to perform a transmission check on the data frame and select a data frame that satisfies the timestamp interval requirement for transmission. The U\_data\_pipe data processing module can realize the control of the data transmission rate and can also solve the problem that different main stations require different transmission rates when the DPDC communicates with the multi-main station.

The  $\mu$ MPMU command pipe (D\_cmd\_pipe) data processing module is responsible for command interaction with the  $\mu$ MPMU to complete the transmission of commands and configuration information, and also write configuration frames to the CFG files. DPDCs with the extended protocol can also send remote commands. The data processing module of the  $\mu$ MPMU file pipe (D\_file\_pipe) is responsible for file command interaction with the  $\mu$ MPMU, sending file commands to the  $\mu$ MPMU, and then receiving and storing the transient data files, which may also be referred to as high dynamic (HD) (the file format refers to [25]). The data processing module of the main station command pipe (U\_cmd\_pipe) is responsible for command interaction with the main station and can read and write CFG files. The data processing module of the main station file pipe (U\_file\_pipe) is responsible for processing the file command of the main station and transmitting the locally stored RT and HD files according to the requirements of the main station. When the DPDC does not have the file requested by the main station, DPDC will send a file command to each  $\mu$ MPMU to find it. If the corresponding file can be found, the file will be transparently transmitted, otherwise, the negative message will be replied to the main station.

The file storage module is implemented by a shell script, and the RT files and the HD files are compressed and stored using the gzip compression program of the Linux system. When the disk capacity reaches the set upper limit, the file storage module deletes the RT file stored for more than 14 days, and also deletes the old HD file and controls the number of HD files to 1000. In addition to the file storage module, DPDC also has some functional modules not shown in Figure 2 implemented by shell scripts, such as watchdog, self-starting, system log and other modules.

#### 3.2. Hardware Platform

DPDC hardware adopts platform hardware architecture design, with high-level electromagnetic compatibility (EMC) protection, its operating system adopts Linux real-time multitasking operating system with independent cutting and packaging. The DPDC front and rear panels are shown in Figure 5. The DPDC model is PDC-2018, with a width of 19 inches and a height of 2 units (U) and a length of 12 inches. U is a unit representing the external dimensions of the server, 1 U = 1.75 inches.



Figure 5. DPDC panel. (a) Front panel; (b) rear panel.

In order to meet DPDC's multi-tasking processing capabilities, the device uses a dual-core dual-threaded CPU, model Intel(R) Celeron(R) 2980 U, with 2 M cache, clocked at 1.60 GHz. On the storage, DPDC needs to store more than 14 days of RT files and more than 1000 HD files. The device uses a 480 GB solid-state drive to meet the requirements of fast reading and writing, low power consumption, shock resistance, and anti-fall. In addition to the space occupied by the system and software, the remaining space of the solid-state drive is greater than 420 GB. The running memory type is DDR3L and the size is 4 GB, which can meet the need to set a large buffer when DPDC accesses a large number of  $\mu$ MPMU. In order to meet the communication with a large number of  $\mu$ MPMU, the DPDC is equipped with six Gigabit adaptive Ethernet ports. The timing interface is RS-485, and InterRange Instrumentation Group Time Code Format B (IRIG-B) is used. The DPDC is connected to a synchronous clock source through this interface and receives a synchronous clock signal. The DPDC supports dual power supply operation and can be powered by direct current or alternating current. Maintenance of the DPDC can be remotely logged in through the network port, or directly connected to the display and keyboard. There are multiple LED indicators on the front panel of the device. The indication information includes the status of the system operation, alarms, timing anomalies, communication anomalies, and data link status.

#### 4. The Test of the Key Performance Indicators of DPDC

In this study, equipment including µMPMU, time source, power signal source, etc., and some software (e.g., Main station Emulator and PMU Emulator) are deployed to build the distribution network WAMS system for DPDC test. The laboratory test environment for DPDC is shown in Figure 6.

The  $\mu$ MPMUs used in the test are produced by XUJI Electrics Company. The model is  $\mu$ M-PMU-851, which can collect three-phase voltages, three-phase currents and zero-sequence current at the same

time and make  $\mu$ MPMUs achieve "three remote" function as the distribution network terminals. The time source is synchronous with GPS and provides time service to  $\mu$ MPMUs and DPDC, manufactured by Xu Ji Group, model DDS100-S, time accuracy  $\leq 1\mu$ S. A relay protection tester works as a power signals source. It is manufactured by Guangdong onlly Electrical Automation Co., Ltd. and its model is ONLLY-AQ430. PC1 has two network ports, Intel(R) Xeon (R) E5-2603 v4 processor, 8 G RAM with 1.7 GHz working frequency. PC2 is a laptop with an Intel (R) Core (TM) i7 6700HQ processor, 4 G RAM with 3.5 GHz working frequency. PMU Emulator and Main station Emulator software are developed for DPDC testing. PMU Emulator is based on Visual C ++ development, supports the standard GB/T 26865.2-2011 protocol, and adopts a multi-thread design. The software can simulate 30 PMUs at the same time, each PMU has three output streams, the data length can be modified; support 1~200 frames/s report rate. The data time of each PMU is initially generated by the computer time. The timestamp can be consistent or a fixed delay can be set to simulate the time difference between different PMUs arriving at the PDC through different channels. Main station Emulator is also developed based on Visual C ++. It can connect multiple PMU and PDC devices and adopt a multi-thread design. The simulated master station displays the received data based on a table. The packet capture software Wireshark is also installed on the DPDC. Its function is to capture network packets and display the most detailed network packet data as possible, with time accurate to microseconds. In the following key indicator tests, PDC Emulator and Main station Emulator are mainly used to test DPDC.



Figure 6. DPDC test environment.

# 4.1. Access Capacity

The number of connected  $\mu$ MPMUs is the basic parameter to the access capacity of DPDC. Meanwhile, the amount of data sent by  $\mu$ MPMU should be considered. The connection diagram of this test is shown in Figure 7. The two network ports of the PC 1 are connected to the two network ports of the DPDC via the network cable 1 and the network cable 2, respectively. The PMU Emulator and the Main station Emulator are run on the PC1, and the simulated PMUs transmit data to the DPDC through the network cable 1. After the DPDC collects the data, the data is transmitted to the simulated main station through the network cable 2.



Figure 7. DPDC access capability test.

The single line measurement configuration of the  $\mu$ MPMU is referenced in the test. Configure each simulated PMU data volume into 12 phasors (three-phase voltage, three-phase current, voltage and current symmetrical components), 20 analog quantities (active power P, reactive power Q, 3rd, 5th, 7th harmonic voltage and current) and 4 digital quantities (external signal lights, relays). A data frame length is 114 bytes. Test at 10, 25, 50, and 100 frame/s reporting rates. Each test increments the number of simulated PMUs, observes whether the data is correct through the simulated master station, and monitors the CPU and RAM utilization of the PDC process through the top command. The test results are shown in Table 3.

Number of Connected PMUs	CPU Utilization (%) at Different Reporting Rates (fps)				RAM Utilization (%)
Number of Connected 1 1105	10	25	50	100	
3	2	5	9	18	0.3
5	3	7	15	29	0.3
10	6	15	22	50	0.5
15	9	20	42	50	0.6
20	11	26	50	50	0.8

Table 3. DPDC access capability test results.

As can be seen from Table 2, as the number of PMUs or the reporting rate increases, the CPU Utilization also increases, and the RAM Utilization is only related to the number of PMUs. In terms of CPU Utilization, the network card driver calls the DMA engine to copy the data packet to the kernel buffer. After the copy is successful, it initiates an interrupt notification interrupt handler. As the number and frequency of input streams increase, message queues trigger the CPU more frequently through interrupts, so CPU Utilization will rise. The DPDC program's internal mechanism also affects CPU Utilization. In terms of RAM Utilization, a buffer of a fixed depth has been allocated according to the configuration file when the DPDC was started. After PMUs are connected, the length of the buffer is also determined by the data length of each PMU. In the absence of other data processing tasks, the RAM Utilization of the DPDC program does not change. The data aggregation of DPDC in this test is normal, and the test result shows that the DPDC meets the access capability requirement.

#### 4.2. Time Difference Tolerance Test

Essentially speaking, the time difference tolerance is equivalent to the multi-channel access capability. In the distribution network, the  $\mu$ MPMUs connected to the same DPDC may use different communication media and routes. The different data packets with the same timestamp would reach DPDC at different times. The laboratory cannot directly reproduce the real situation of the power line network and the wireless network, so we use the PMU Emulator for further testing. The connection diagram of this test is shown in Figure 8; the DPDC directly connects to the PC1 and PC2 via a network cable. The Main station Emulator runs on the PC1, and the PMU Emulator runs on the PC2.



Figure 8. DPDC time difference tolerance test.

Eight PMUs were simulated on the PMU Emulator, and different delays for the simulated PMUs to test were set. The eight simulated PMUs are numbered sequentially and then divided into two groups. The odd-numbered PMU does not set the delay, and the even-numbered PMU sets the same delay to simulate the delay in the actual channel. After DPDC establishes communication connections with each simulated PMU, the simulated main station summons the DPDC data and observe whether the DPDC can correctly aggregate and forward PMUs data frames of different delays. Delays of 50, 100, 150, 200, and 300 ms were set respectively to test, and PMUs' reporting rate was 100 fps. Table 4 shows the results.

The test results showed that when the time difference between the simulated PMU data is within 200 ms, the DPDC can correctly aggregate the data, and when the time difference reaches 300 ms, the DPDC aggregate data is abnormal. When data aggregation is abnormal, the DPDC aggregates data arrives at different times separately and outputs two data streams, one of which contains PMU data arriving earlier and the other containing PMU data arriving later. It can be observed from the Main station Emulator that the reporting rate of DPDC is increased to 200 fps (twice the normal value), and each data frame contains only half the number of PMU data after parsing. The cause of the abnormality is that the time difference between the PMU data exceeds the depth of the DPDC buffer. When the data that arrives earlier fill each layer of the buffer, the continued arrival of data prompted DPDC to push the data at the top of the buffer away, and the data that arrives later has no opportunity to aggregate with the data that has been pushed away. When the data aggregation is normal, the increase in PMU data delay is positively related to the delay of PMU data arriving later. When the data aggregation is abnormal, the increase in PMU data delay is the depth of the DPDC buffer. The reason why there is no change in CPU and RAM Utilization is the same as the analysis in Section 4.1. With a fixed number of PMUs and reporting rate, if there are no other data processing tasks, CPU and RAM Utilization will not change significantly. According to the test results, the time difference tolerance of DPDC reaches the requirement. When the time difference between  $\mu$ MPMUs connected through different channels is within 200 ms, DPDC can correctly aggregate data.

PMU Delay (ms)	CPU Utilization (%)	RAM Utilization (%)	Aggregation
50	45	0.4	Normal
100	45	0.4	Normal
150	45	0.4	Normal
200	45	0.4	Normal
300	45	0.4	Abnormal

Table 4. DPDC time difference tolerance test results.

#### 4.3. Data Delay Test

Measuring the PDC data delay in the transmission network WAMS requires the network tester, but the network tester is expensive and complicated to operate. In this study, a simple way to measure the data delay of the DPDC is proposed. In order to ensure that the data transmission time and the data reception time have the same benchmark, the packet capture software Wireshark runs on the DPDC. Wireshark records the time when the last frame of the data frame of the same timestamp of each simulated PMU arrives at the DPDC and the time when the DPDC sends the data frame with the timestamp and calculates the difference between the two times to obtain the corresponding data delay. The test connection diagram can choose one of the above two schemes. During the test, the time differences over a period are averaged as the DPDC data delay. The number of PMUs is initially set to one and then set to two, followed by two increments each time. PMUs report rate is set to 100 fps. Figure 9 shows the test results.



Figure 9. DPDC data delay test.

The test results show that the data delay of the designed DPDC increases with the number of accessing PMUs. When 20 PMUs are connected, the data delay remains within 1 ms, which meets the test requirement.

## 5. Actual Installation and Operation

The designed DPDC is currently located in the Lingang area of Pudong New Area, Shanghai, China. The area has a full-voltage distribution network of 220 kV or less. The distribution network is characterized by typical urban power grids. The feeder lines are cable lines or cable/overhead hybrid lines. It includes substations of various voltage levels, multiple electric vehicles charging stations, multiple photovoltaic power stations, and wind power plants. The DPDCs and  $\mu$ MPMUs have been installed in some 220 kV substations, 35 kV substations, 10 kV switchyards, and photovoltaic grid points. The distribution network WAMS constructed by the project has functions such as distribution network state estimation, fault diagnosis and location, and island control. Figure 10 shows some actual installation scenes.



Figure 10. Field installation. (a) The front layout of the cabinet; (b) a µMPMU mounted on the column.

(a) Is the installation of equipment in a 35 kV substation, 7  $\mu$ MPMUs and one DPDC are assembled in two cabinets. (b) Shows a  $\mu$ MPMU installed on the feeder, which is placed in a dedicated box. The information of the line is measured by the voltage and current transformers and the data of the  $\mu$ MPMU is sent to the substation. The WAMS main station of the distribution network is deployed in the Shanghai Pudong Power Supply Company of the State Grid. The D5000-WAMS front-end machine developed by the NARI Group can view real-time data.

The designed DPDC has been running for several months. During the time, DPDC works stably and the data transmission is normal. The project is still in progress and more DPDCs and  $\mu$ MPMUs will be deployed in the future.

#### 6. Conclusions

With the distributed energy, flexible load and electric vehicle charging station (pile) connected to the distribution network on a large scale, the monitoring and control methods of the traditional distribution network have difficulty coping with the new problems faced by the distribution network. Therefore, the synchronous phasor measurement technology in the transmission network WAMS is introduced into the distribution network, and the  $\mu$ MPMUs with the power distribution terminal functions are gradually deployed to each key node in the distribution network. The DPDC for the distribution network communication environment has become more important as a connection node between the  $\mu$ MPMUs and the main station. In view of the current lack of discussion on the design and application of the DPDC, this paper analyzes and compares the DPDC and the traditional PDC in terms of functions, communication methods, and application requirements. The key indicators for evaluating the performance of DPDC are proposed, and a DPDC is designed. The test environment is further designed to perform various evaluations on the DPDC, and the evaluation results are analyzed. The contribution of this paper is summarized below:

- This paper introduces three basic networking structures of μMPMU-DPDC and illustrates the various communication methods that may exist in the networking. Then, by comparing the PMU and μMPMU, PDC and DPDC from the installation location and main functions, the application requirements of DPDC are obtained. Further to judge the basic performance of DPDC, the key performance indicators of DPDC are provided, namely, the access capability is no less than 20 μMPMU, the time difference tolerance is no less than 150 ms and the data delay is within 3 ms;
- A design method of hardware DPDC is proposed, which uses event-driven mechanism and structured program design, uses libuv to establish TCP connection, and completes the asynchronous non-blocking operation for multi-task through the multi-thread and callback mechanisms. The functions of each software module inside DPDC are introduced, and the data buffer structure and data aggregation mechanism are described in detail. The data aggregation mechanism and data structure ensure the efficiency of data processing and reduce the demand for hardware resources. The hardware configuration and selection of the DPDC are also described;
- The test methods for key performance indicators of DPDC are proposed. Firstly, the development of the test environment and the test software are described. Then test method for key performance indicators and conduct various tests on the DPDC was designed. The test results show that the designed DPDC access capability can reach 20 µMPMUs, the time difference tolerance is not less than 200 ms, and the data delay is less than 1 ms, which meets the requirements of all key performance indicators. Proven that the performance of the designed DPDC meets its application in the distribution network. Further analysis of the test results shows that the main impact on the performance of DPDC is the program mechanism and hardware performance. The designed DPDC uses a multi-threaded and event-driven mechanism. DPDC will call the corresponding callback function for processing when it receives the data. When the data aggregation is completed, it will be sent to the master station immediately to achieve a small delay and the CPU utilization rate is not high. If the time-driven mechanism is used to periodically query the status of each socket in the receiving thread, the DPDC data delay and CPU utilization rate will increase accordingly.

The field application of the designed DPDC works well. The further work of the author's team is to expand more distribution sub-station functions and islanded control functions on the DPDC, enabling the DPDC to act as a distributed control sub-station for distribution automation and DER grid-connected control tasks.

**Author Contributions:** Conceptualization, W.T. and M.M.; methodology, W.T. and M.M.; software, validation, M.M., D.X. and Y.S.; writing—original draft preparation, M.M.; writing—review and editing, M.M., W.T. and M.D.; resources, W.X. and C.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by The National Key R & D Plan of China, grant number 2017YFB0902800; State Grid Corporation Science and Technology Project, grant number 52094017003D; the Science and Technology Research Project of Anhui Province, grant number 1704a0902004.

Conflicts of Interest: The authors declare no conflict of interest.

## References

- 1. Cruz, M.A.R.S.; Rocha, H.R.O.; Paiva, M.H.M. An algorithm for cost optimization of pmu and communication infrastructure in WAMS. *Int. J. Electr. Power Energy Syst.* **2019**, *106*, 96–104. [CrossRef]
- Jamil, E.; Rihan, M.; Anees, M.A. Towards optimal placement of phasor measurement units for smart distribution systems. In Proceedings of the 2014 6th IEEE Power India International Conference (PIICON), Delhi, India, 5–7 December 2014; pp. 1–6.
- 3. Tao, W.; Ma, M.; Ding, M.; Xie, W.; Fang, C. A Priority-Based Synchronous Phasor Transmission Protocol Extension Method for the Active Distribution Network. *Appl. Sci.* **2019**, *9*, 2135. [CrossRef]
- 4. Von Meier, A.; Stewart, E.; Mceachern, A.; Andersen, M.; Mehrmanesh, L. Precision micro-synchrophasors for distribution systems: A summary of applications. *IEEE Trans. Smart Grid* **2017**, *8*, 2926–2936. [CrossRef]
- 5. Yang, X.; Zhang, X.P.; Zhou, S. Coordinated algorithms for distributed state estimation with synchronized phasor measurements. *Appl. Energy* **2012**, *96*, 253–260. [CrossRef]
- 6. De, L.R.J.; Centeno, V.; Thorp, J.S. Synchronized phasor measurement applications in power systems. *IEEE Trans. Smart Grid* **2010**, *1*, 20–27.
- 7. Aminifar, F.; Fotuhi-Firuzabad, M.; Safdarian, A.; Davoudi, A. Synchrophasor measurement technology in power systems: Panorama and state-of-the-art. *IEEE Access* **2014**, *2*, 1607–1628. [CrossRef]
- 8. IEEE Standard C37.244–2013. IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring; IEEE: Piscataway, NJ, USA, 2013.
- 9. IEEE C37.247-2019. *IEEE Standard for Phasor Data Concentrators for Power Systems*; IEEE: Piscataway, NJ, USA, 2019.
- 10. Derviskadic, A.; Romano, P.; Pignati, M.; Paolone, M. Architecture and experimental validation of a low-latency phasor data concentrator. *IEEE Trans. Smart Grid* **2016**, *8*, 2885–2893.
- 11. Armenia, A.; Chow, J.H. A flexible phasor data concentrator design leveraging existing software technologies. *Ieee Trans. Smart Grid* **2010**, *1*, 73–81. [CrossRef]
- Adamiak, M.G.; Kanabar, M.; Rodriquez, J.; Zadeh, M.D. Design and implementation of a synchrophasor data concentrator. In Proceedings of the 2011 IEEE PES Conference on Innovative Smart Grid—Middle East, Jeddah, Saudi Arabia, 17–20 December 2011; pp. 1–5.
- Pourramezan, R.; Seyedi, Y.; Karimi, H.; Zhu, G.; Mont-Briant, M. Design of an advanced phasor data concentrator for monitoring of distributed energy resources in smart microgrids. *Ieee Trans. Ind. Inform.* 2017, 13, 3027–3036. [CrossRef]
- 14. Castello, P.; Muscas, C.; Pegoraro, P.A.; Sulis, S. Active Phasor Data Concentrator performing adaptive management of latency. *Sustain. Energygrids Netw.* **2018**, *16*, 270–277. [CrossRef]
- Castello, P.; Muscas, C.; Pegoraro, P.A.; Sulis, S. Low-Cost Implementation of an Active Phasor Data Concentrator for Smart Grid. In Proceedings of the 2018 Workshop on Metrology for Industry 4.0 and IoT, Brescia, Italy, 16–18 April 2018; pp. 78–82.
- Liu, Y.; Zhan, L.; Zhang, Y.; Markham, P.N.; Zhou, D.; Guo, J.; Lei, Y.; Kou, G.; Yao, W.; Chai, J.; et al. Wide-Area Measurement System Development at the Distribution Level: An FNET/GridEye Example. *IEEE Trans. Power Delivery* 2015, *31*, 721–731. [CrossRef]

- Kanabar, M.; Adamiak, M.G.; Rodrigues, J. Optimizing Wide Area Measurement System architectures with advancements in Phasor Data Concentrators (PDCs). In Proceedings of the 2013 IEEE Power & Energy Society General Meeting, Vancouver, BC, Canada, 21–25 July 2013; pp. 1–5.
- Gore, R.N.; Valsan, S.P. Wireless communication technologies for smart grid (WAMS) deployment. In Proceedings of the 2018 IEEE International Conference on Industrial Technology (ICIT), Lyon, France, 20–22 February 2018; pp. 1326–1331.
- 19. Kansal, P.; Bose, A. Bandwidth and Latency Requirements for Smart Transmission Grid Applications. *IEEE Trans. Smart Grid* **2012**, *3*, 1344–1352. [CrossRef]
- 20. IEEE Standard C37.118-2011.2. *IEEE Standard for Synchrophasors Data Transfer for Power System*; IEEE: Piscataway, NJ, USA, 2011.
- 21. Field Test Asia Pte Ltd. *GB/T26865.2-2011 Real-Time Dynamic Monitoring Systems of Power System-Part 2: Protocols for Data Transferring;* Field Test Asia Pte Ltd.: Singapore, 2011.
- 22. Tao, W.; Zhu, X.; Fang, C.; Liu, J.S. Research on Main Indicators and Test Methods of Distribution Network Phasor Concentrator. *Power Syst. Technol.* **2019**, *43*, 801–809.
- 23. iPDC—Free Phasor Data Concentrator—CodePlex Archive. Available online: https://archive.codeplex.com/ ?p=ipdc (accessed on 4 January 2018).
- 24. Welcome to the Libuv Documentation. Available online: http://docs.libuv.org/en/v1.x/index.html# (accessed on 1 May 2018).
- 25. IEEE Standard C37.111-1999. IEEE Standard for Common Format for Transient Data Exchange (COMTRADE) for Power Systems; IEEE: Piscataway, NJ, USA, 1999.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).