Active Phasor Data Concentrator Performing Adaptive Management of Latency

Paolo Castello, Carlo Muscas, Paolo Attilio Pegoraro, Sara Sulis
Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy.
[paolo.castello, carlo, paolo.pegoraro, sara.sulis]@diee.unica.it

Abstract— **The Phasor Data Concentrator (PDC) is a function in charge to receive and combine the time-tagged synchrophasor data from Phasor Measurement Units (PMUs). The tasks of the PDC can include data handling, processing, and storage. Collected data are forwarded to the next higher-level element of the hierarchical monitoring architecture, which means either an operational center or a higher level PDC. Definitions of the terminology, functional descriptions and the test procedures concerning the PDC can be found in the guide IEEE C37.244-2013. In particular, in order for the PDC to ensure good latency performance, its interfacing with both the lower hierarchical level (i.e. PMUs with different features) and the higher one must be done in a reasonable time. It is worth noting that, while PMUs and PDCs were originally conceived for transmission systems, they are now expected to become key elements also for the monitoring of modern distribution grids. In this evolving and complex scenario, the PDC could play a crucial and active role. In this paper, an active PDC with advanced functionalities is proposed to manage the delay of several PMU streams so that an original adaptive data aggregation policy is implemented to allow compliance with time constraints of real-time applications.**

Index Terms* —*Phasor Data Concentrator, Phasor Measurement Unit; Latency; Power Systems; Smart Grid.

1. INTRODUCTION

Modern Wide Area Measurement Systems (WAMSs) for power grids exploit the benefits offered by the synchrophasor technology and made available through its main components: Phasor Measurement Units (PMUs) and Phasor Data Concentrators (PDCs). The PMU, which can be seen as the base "sensor" of the WAMS, measures voltage and current synchrophasors, as well as frequency and Rate Of Change Of Frequency (ROCOF) [1],[2]. The PDC represents instead a kind of communication node: it receives the measured data from PMUs and processes them, by aligning the data with the same timestamp to define a single time-synchronized output data stream [3],[4].

In transmission networks, the applications based on synchrophasor measurement systems include state estimation, voltage stability assessment, fault location, line parameters identification, post-mortem analysis, and power system restoration [5]. More recently, the possibility of exploiting synchrophasor technology also in modern distribution networks has been studied [6], and several pilot projects are being developed to this purpose (see for example [7] and [8]).

In this scenario, the number of installed PMUs and PDCs is increasing, thus leading to the need to manage properly the consequent increasing flow of synchronized data.

From this standpoint, the performance of the monitoring system strictly depends on the supporting communication infrastructure. Generally, especially in the case of distribution grids, this infrastructure is not dedicated to the synchrophasor data, but is shared with other applications [9], [10]. Appropriately designed architecture solutions based on the monitoring of Quality of Service (QoS) performance can improve the transmission through a wide area communication network [11], [12].

Synchrophasor systems can be subjected to communication delays and packet dropout that can damage data collection and, as a consequence, compromise advanced control and monitoring functions. A major issue can be represented, for instance, by the presence of data burst that can saturate the network, causing an increase in latency or data loss. Moreover, the different distance between the single PMUs and the PDC implies possible different latencies for individual streams. In [13] this fact is underlined for transmission systems, but the same holds also for distribution systems. Furthermore, the presence of redundant paths, which increase the reliability of the communication system, could cause additional variability to the latency of the PMU streams.

In its classical implementation, during any type of operating conditions, before sending an output data stream the PDC has to wait for all the data sent by all the connected PMUs with the same timestamp. If a stream does not arrive by a given time limit, it is discarded and the PDC sends the (incomplete) set of the available data. The value of this time limit should be properly chosen, according to the requirements of the application [14].

Recently, research activity has been focused on issues related to PDC design, implementation and performance. In [15], a review on existing literature in the field of PDC design and performance assessment is reported and an architecture is proposed for a PDC that implements both the absolute and relative time data pushing logics together with a third one that aims at minimizing the latency introduced by the PDC without increasing the data incompleteness, as suggested in [4]. In [16], focusing on transmission systems, a PDC with adaptive waiting time is proposed to reduce the frequency of adapting control gain in the delay-robust damping control system. In [17] an advanced PDC, designed and implemented on a real-time software platform, is proposed with the aim of enabling the monitoring of DERs in smart micrograds. The proposed PDC is composed of a conventional PDC, a compensation unit, and a monitoring unit applying an adaptive compensation scheme to achieve an estimate of missing data elements.

In this context, in [18] the authors have presented the first step of a research activity aimed at designing an active and flexible PDC with a simple latency management policy. The PDC design proposed in [18] monitors, on a statistical basis, the delay of several input streams and modifies the data aggregation with a plain decisional rule. The technique was validated in a simulated scenario where the PMUs have the same design and configuration.

To overcome some limitations of the approach presented in [18], in this paper, a new solution for a smarter active PDC is presented and a full validation in realistic scenarios is provided. In particular, the policy for the intelligent management of latency, depending on actual operating conditions, has been completely re-designed to make it able to evaluate the latency of each single incoming stream and to adapt its data processing and communication, as well as the waiting time, to meet latency constraints given by specific applications. At the same time, the statistical monitoring function is used to change adaptively the decisional delay threshold for each PMU.

To validate the approach, the policies are tested by means of both simulated and experimental data in a realistic scenario. The newly designed PDC is implemented and validated by means of an appropriately designed test setup where the proposed PDC is connected to commercial PMUs representing classes P and M and realistic network impairments can be emulated.

2. LATENCY IN SYNCHROPHASOR MEASUREMENT SYSTEM

Fig. 1 shows a schematic representation of a synchrophasor measurement system, aimed at highlighting the main contributions to the overall latency:

a) PMU reporting latency

The PMU reporting latency is the difference between the timestamp associated to each measured value and the instant in which the relevant packet is ready to leave at the PMU output (first bit of the outgoing packet). The major contribution to this latency is usually due to the duration of the acquisition window required by the measurement algorithm [19]. Indeed, every PMU compliant with [1] and [2] is characterized by one of the two standardized performance classes: the P class, designed for protection applications that need fast response and low latency, or the M class, aimed at monitoring applications, which is characterized by higher latency due to the longer acquisition windows required for better rejection of some disturbances [20]. As an example, considering fifty measurements per second, P-class devices commonly operate on at least two periods of the signal at the nominal frequency, while M-class PMUs work on six periods.

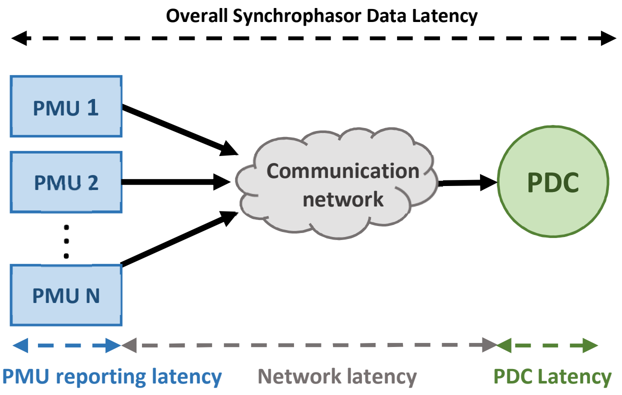

Fig. 1. The main contributors of the overall latency in a synchrophasor measurement system.

b) Network latency

The communication channel is generally shared with other applications that influence the network latency, which could be significantly variable, in particular if the communication channel is public. Moreover, a possible reconfiguration of the network topology implies changes in the communication path, thus affecting the time in which the streams reach the PDC.

c) PDC latency

The PDC latency highly depends on the functionalities implemented in the PDC (see [15] as an example). This contribution is usually much lower than the previous ones.

3. PROPOSED ADAPTIVE MANAGEMENT OF LATENCY

The founding point of the proposed policy is that an effective management of the latency should be based on the requirements of the possible target applications (e.g., response time between less than 0.1 s and tens of seconds can be required by different wide area protection applications, [21]) and should dynamically depend on the actual operating conditions of the overall system.

In this context, a management policy with different decisional levels has been designed: the PDC collects information on available PMUs and target applications, monitors the latencies of the input streams, detects possible changes, and, consequently, adapts the aggregation of data in the PDC output stream through a suitable strategy.

In a synchrophasor system, this active functionality allows adapting data processing and communication to handle simultaneously different types of PMUs and to meet latency constraints given by specific applications.

In the following, the proposed procedure is described in detail.

As a preliminary step, the time limits characterizing the procedure have to be set. In this paper two main time parameters have been considered: t_L, , that is the time limit depending on the target real-time applications, and t_{out}, that is the timeout limit (maximum waiting time after which any packet is considered lost, usually set in the order of seconds) that preserves data for other interested applications. The threshold limits are inputs of the proposed PDC, which is conceived as a flexible device, able to work with different possible thresholds that, in any case, have to be provided by the system operators. These parameters can be set by the operators depending on specific application needs and on their specific architectures and constraints (as an example, a discussion on latency needs in the context of voltage regulation in distribution systems can be found in [22]). In the proposed implementation (see Section 4) the time parameters are thus considered as configuration data.

As discussed in Section 2, the time delay $t_{PDC\ input_i}$ after which a packet of the *i*-th input stream arrives at the PDC input depends on two main contributions:

$$t_{PDC\ input_i} = (t_{PMU\ RL_i} + t_{NL_i}) \tag{1}$$

where $t_{PMU\ RL_i}$ is the PMU reporting latency, while t_{NL_i} is the corresponding network latency.

Equipping a PDC with a suitable time synchronization with respect to an UTC source allows measuring latency. This characteristic is not mandatory for a commercial PDC (see Section 5 in [4]), but it is an essential feature to implement the proposed approach and to assess the communication network behavior. The absolute arrival time T_{arrive_i} of a packet of the *i*-th stream can be accurately evaluated. Thus, since the absolute timestamp of the measurements, $T_{timestamp}$, is reported in the PMU packet, the time delay $t_{PDC\ input_i}$ can be simply assessed by extending the concepts described in [19] for the PMU latency:

$$t_{PDC\ input_i} = T_{arrive_i} - T_{timestamp} \tag{2}$$

Based on these single values, the latency that characterizes the streams coming from each PMU can be represented by means of a statistical representation [18]:

$$t_{s_i} = \mu_i + 3\sigma_i \tag{3}$$

where μ_i is the mean value of the latency of the packets of the *i*-th PMU and σ_i is its standard deviation. It is worth noting that (3) is only an example of possible evaluation of t_{s_i} and other statistics could be computed and applied, depending on the considered PMU scenario.

Thus, once both the single specific value of the latency and its statistical monitor function are available for all the input streams, the decisional logic for the aggregation of the output packets can be implemented.

The main idea is that, in order to keep latency performance as required by real-time applications, the PDC must always send to the next hierarchic level by the given time limit t_L a main output stream (critical stream) that contains all the data of the streams that have a low stream latency (relying on (3) for the distinction) which are available by that limit. Then, the PDC will send in separate and delayed output stream(s) the possible late input(s) that can be useful for the other applications, which do not have strict latency limits or do not need to operate in real-time. It is important to highlight that the aim is to preserve stream latencies as far as possible. A critical application must rely on timely data and the proposed approach prevents this application from unnecessarily waiting (and thus, practically, losing) all the data because of a single late stream.

It is worth noting that, in any case, the PDC cannot wait indefinitely late streams from the PMUs and, thus, after a reasonable timeout, t_{out}, the data are considered as lost.

For a smarter management of the size of the output streams and, consequently, of the communication bandwidth, the situations in which late streams occur must be further differentiated into two cases:

i) a late packet is an outlier for a PMU stream whose latency is statistically low;
ii) all the packets coming from a PMU are delayed, thus outlining a possible issue in the corresponding communication channel.

On this basis, the following policy can be adopted for every incoming packet of each stream:

Case 1) If the statistical latency of all streams is below the limit *($t_{s_i} < t_L$),* the following three scenarios are of interest:

a) Normal condition:

$$t_{PDC\ input_i} < t_L \tag{4}$$

The PDC output stream is composed by all the data provided by the PMUs with the same timestamp;

b) Delay condition:

$$\exists\, i \mid t_L \leq t_{PDC\ input_i} < t_{out} \tag{5}$$

A single data packet of the *i*-th PMU arrives late with respect to the time limit t_L. The PDC output stream is thus sent without the data of this late packet. Such *i*-th stream packet is then collected for a follow-up low priority output. This event is considered as an outlier and is not representative of an actual variation of the trend of the *i*-th PMU latency;

c) Timeout condition:

$$\exists\, i \mid t_{PDC\ input_i} \geq t_{out} \tag{6}$$

In this case, the packet of the i-th stream is considered too late for any target application and is discarded.

Case 2) If the statistical latency exceeds the limit for at least one stream $(\exists\, i \mid t_{s_i} \geq t_L)$, the i-th stream has to be separated from the others and forwarded into a different PDC less-critical output stream (with looser latency constraints). In this case, a new, higher, time limit nt_L is defined for that PMU, whose data will be then subjected to similar rules as before, by replacing t_L, with nt_L in (4) and (5).

If the behavior of that PMU still worsens and becomes $t_{s_i} \geq n\mathrm{t_L}$, the limit can be further increased to $(n+1)t_L$, and so on until the timeout value is reached.

It is important to highlight that each limit of nt_L is only used as the maximum waiting time for the corresponding delayed output PDC stream and is conceived as it is typically done for different QoS classes to give latency guarantees for worst cases. For example, considering the first latency range, if a stream (PMU i) undergoes only a small latency increase above the limit t_L, the packets of the delayed output stream is sent when all the packets of PMU i are ready, without waiting the upper limit $2t_L$. The aim is always to preserve latency as far as possible.

4. TESTS SCENARIOS

The proposed methodology has been validated by means of an appropriate test setup and two scenarios.

4.1 Homogeneous PMU scenario

In the first scenario, called the "homogeneous" scenario, a simulation setting was developed in LabVIEW environment. The aim is to study the behavior of the system in case of data latency provided by real PMU measurements stored in a public database [23]. The latency values correspond to a pilot smart grid monitored with four identical PMUs (PMU 1A to PMU 4A) compliant with the P class. The PMUs are configured to send the IEEE C37.118.2 data frame in an UDP packet every 20 ms (reporting rate equal to 50 frames/s, hereafter indicated as 50 fps). Packets are collected by a PDC in a dedicated communication network. Table I shows the average and standard deviations of the packet latencies from the four PMUs for one-hour data record. These values are very stable over the database, thus a portion of six-minute of this data record (representative of the entire data set) is used for the simulation tests in Section 5.1. The aim of these preliminary investigations is to validate the policy with controlled impairments.

TABLE I. MEAN VALUES AND STANDARD DEVIATIONS OF THE LATENCY OF THE PDC INPUT STREAMS – P-CLASS PMUS - HOMOGENEOUS SCENARIO

PMU	Mean Value [ms]	Standard Deviation [ms]
1A	41.1	0.54
2A	44.0	0.56
3A	44.0	0.56
4A	46.3	0.57

4.2 Heterogeneous PMU scenario

A second "heterogeneous" scenario has been experimentally designed in the measurement laboratory at University of Cagliari, considering commercial PMUs with different characteristics, to test the proposed procedure in another realistic context. In fact, in a WAMS context, the settings and features of PMUs installed in the field can be very different. In this regard, one of the main goal of the synchrophasor measurement standardization process, with its published standards, version updates and integrations, has always been to guarantee the interoperability between PMUs from different vendors. Nevertheless, new compliance limits and latency requirements have been introduced in the last version of the standard for different performance classes. Focusing on the latency, the definition of PMU reporting latency was introduced for the first time in the 2011 [1]. Therefore, PMUs built before [1] have different latency performance, as shown in [19] for a PMU compliant with the 2005 version.

Table II summarizes the main characteristics of four different commercial PMUs (PMU 1B to PMU 4B) typically used to monitor transmission systems and configured to send 50 fps. Each PMU has different specifications and is compliant with a different issue or version of the synchrophasor standard. In particular, PMU 1B does not specify its compliance information, being the year of manufacture the only available information.

TABLE II. MAIN CHARACTERISTICS OF COMMERCIAL PMUS - HETEROGENEOUS SCENARIO

PMU	Version of the standard	Level or Class of Performance	Data Packet size [bytes]
1B	Not specified (built in 2015)	Not specified	108
2B	2005	1	90
3B	2014	M	72
4B	2011	M	138

The PMUs are configured for a TCP communication and the algorithm chosen for the test setup, where possible, is an algorithm declared as compliant with "M class" (standard versions 2011 and 2014) or "1 level" (standard version 2005).

In addition, the size of the data packet is different for each PMU, as it depends on the amount of information and type of data included in the packet (number of phasors, fixed 16-bit or floating-point format, digital and analog channels, etc…).

The PMUs are connected to the PDC using the scheme in Fig. 2. The PMUs are directly connected to a high-performance switch, built for energy communication infrastructure. To create different network scenarios, a network emulator is introduced in the architecture. The network emulator is a workstation (Intel E8500, 8 GB RAM), running Linux O.S. (Ubuntu), set as a transparent bridge and running software NetEm. The workstation is equipped with two ethernet network interface cards to manage the input/output streams. In a normal operation, when no impairments are configured, the system can be seen as a transparent bridge between the PMU and the PDC and the added latency is negligible. For the tests, the connections depicted in Fig. 2 are adopted.

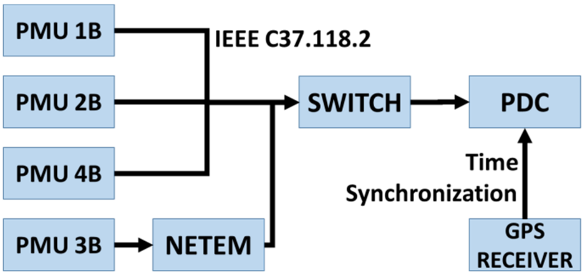

Fig. 2. Test scenario with heterogeneous PMUs and network emulator.

In the following, PDC functionality runs on a workstation (Intel E5645, 6 GB RAM) equipped with Windows 10 O.S. and able to manage the analyzed four PMU streams in input and to time-tag the data packets. It is important to highlight that the PDC prototype used for the tests serves as a proof-of-concept for the implementation and validation of the proposed stream latency management strategies and is not intended for direct implementation of a commercial PDC.

The time management of the proposed PDC is based on the evaluation and correction of the offset between the time obtained by Network Time Protocol (NTP) synchronizing the workstation clock and the Pulse-Per-Second (PPS) signal obtained from the GPS receiver (a Symmetricom XL-750 GPS time source with a 100 ns accuracy) acquired by means of a NI USB-6211 data acquisition board. To keep the relationship between the NTP time and the PPS signal, a software decoder of unmodulated DC level shift (DCLS) IRIG-B signal, acquired with PPS triggering every second, provides the full date and time once per second. The time offset is thus evaluated every second and corrected in the latency evaluation process. The time offset has been characterized and the standard deviation obtained in 1 hour is equal to 0.6 ms and, thus, is sufficiently accurate for the purpose. In this way, the PDC receives the streams and collects, aligns and eventually forwards the data to a higher level PDC.

In the proposed implementation, the time parameters t_L and t_{out}, can be read from a configuration file (or web service) or automatically configured via a user-defined C37.118.2 command frame, depending on the target application latency needs.

TABLE III. MEAN VALUES AND STANDARD DEVIATIONS OF THE LATENCY OF THE PDC INPUT STREAMS – COMMERCIAL PMUS - HETEROGENEOUS SCENARIO

PMU	Mean Value [ms]	Standard Deviation [ms]
1B	53.1	8.30
2B	16.3	0.72
3B	109.8	0.94
4B	98.0	1.00

Table III reports the averages and the standard deviations of the latency values measured by the PDC when no impairments or contingencies are introduced by the network emulator. As it is clear from the values, a high variability in latency is a realistic scenario for an actual WAMS implemented with different commercial PMUs, and PMU reporting latency can also vary significantly depending on implementation.

It can be useful to discuss briefly how the stream management can be performed in practice. In this paper, the focus is on the scientific proposal for handling different streams from a latency viewpoint and, thus, currently available protocols should not be here considered as a strict constraint. In particular, protocol C37.118.2 is conceived for backward compatibility and is not entirely suitable for an efficient implementation of the proposed policies. For these reasons, the proposed PDC has been implemented by extending the C37.118.2 functionalities, which can be done using either user-defined fields or sending unsolicited configuration frames (in [4] such possibility is contemplated). It has to be observed that implementations compliant to C37.118.2 could be also defined (e.g. configuring two communication channels alive and keeping one of them silent when unnecessary), but they would be clearly suboptimal from a network bandwidth point of view.

In a desirable extended scenario, protocols will allow more efficient implementations. Actually, the guide [4] indicates as desirable a revision of C37.118.2 for advanced features implementation and also underlines that many functionalities are not covered by current standards, recommending manufacturers' agreements for their implementation. Of course, the application relying on the active PDC has to be aware of the protocol, since the proposed PDC aims at addressing its specific needs, but this is the basis of every high-level monitoring system dealing with multiple data sources.

5. TESTS AND RESULTS

Both normal and changing operating conditions have been tested and analyzed for the two scenarios described in Section 4. In the following, the most significant results are presented and discussed.

In the tests, the computation of the average and standard deviation in monitor functions (3) is based on a five-second window, which includes 250 packets, since the reporting rate of the PMUs is set to 50 fps. When there are missing packets, their latency is artificially set as the difference between the current time and their (missing) timestamp, with an upper bound equal to t_{out}.

5.1 Homogeneous PMU scenario

Fig. 3 presents the values of latency of the synchrophasor data provided from the four equal PMUs.

The devices have the same reporting latency. Thus, the different latency values are caused by the diverse network paths from each PMU to the PDC.

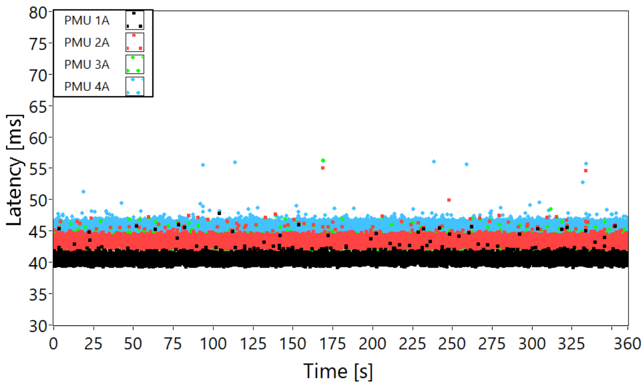

Fig. 3. PDC input data streams provided by the four PMUs.

The monitor functions (3) corresponding to a normal condition are shown in Fig. 4, where $Max\ ts$ represents the maximum value of such functions. In this case, all the streams are under the limit chosen for this scenario (t_L= 50 ms, [24], [25]).

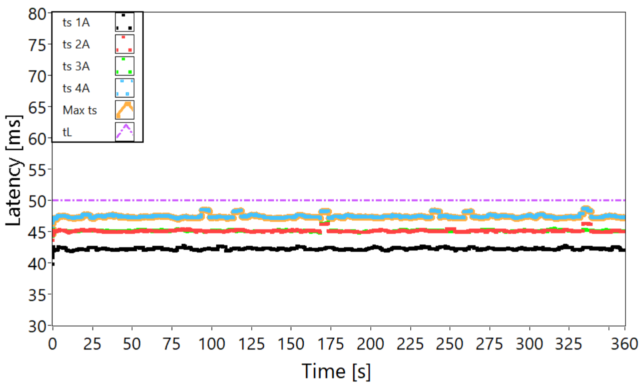

Fig. 4. Trend of monitor functions (ts) and the chosen limit (t_L).

Fig. 5 presents the output data, considering the PDC processing time to be negligible. The data received with a latency value over the limit imposed are stored and sent in a cumulative data packet with low priority for non-real-time applications.

While in most cases all the single latency values are below the limit (case 1*a* in Section 2), in some circumstances the data from one PMU arrives later (case 1*b*). In this case, without the latency policy, the PDC would wait for the arrival of the delayed data packets, thus delaying the outbound data (tPDC output w.o. ML in Fig. 5). With the proposed management of latency, the output data (tPDC output with ML in Fig. 5) are sent no later than the time instant set depending on the imposed t_L, while the few late packets of each single stream are stored and kept for the purpose of possible off-line applications. They can be aggregated in a further stream to be sent when needed or requested. This can be performed in several ways, but specific details are implementation dependent and are outside the scope of this paper.

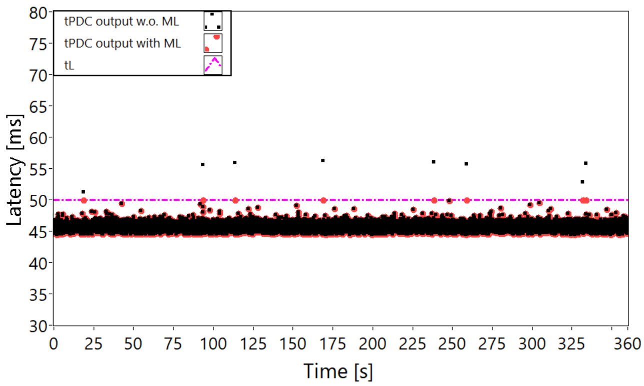

Fig. 5. Output data streams without latency management (tPDC output w.o. ML) and with latency management (tPDC output with ML).

To simulate a possible network issue, a constant value of 20 ms was added to the data of the latency of PMU 3A starting from second 100, as shown in Fig. 6.

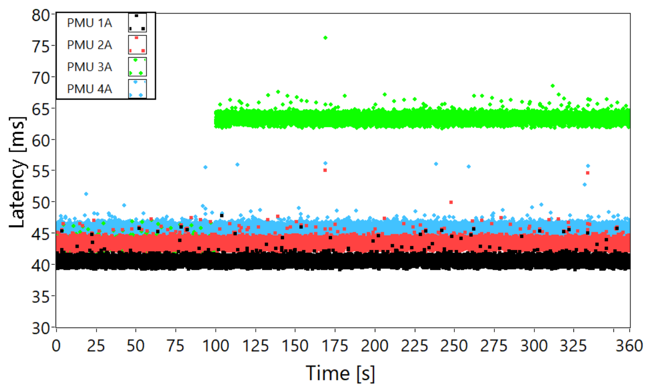

Fig. 6. Data provided by the PMUs in the presence of a simulated variation of the latency conditions for stream 3A.

Fig. 7 reports an x-axis zoomed version of the latency graphs (2 s interval) to better show the policy mechanism around the event. It thus shows the corresponding trend of the monitor functions (3) referring to the different input streams. The stream of PMU 3A exceeds the imposed limit t_L, thus giving rise to the case 2 described in Section 3. According to the policy, the PDC output streams are divided to meet the constraints imposed. It is worth noting that in Fig, 7, as well as in the following Fig. 13, $Max\ ts$ shows the maximum value of the monitor functions for the input streams that are included in the first, non-delayed, output stream.

The detection of the event that leads to stream separation is fast in this case, but its promptness depends on the computation interval chosen for the monitor function. A similar effect occurs when streams have to be merged again because the monitor function of the late stream(s) returns below the limit. Different strategies could be also easily implemented for the two opposite events.

In any case, in order to highlight the advantages of the approach, it is important to compare these small transition intervals with the possibly large event durations (network or PMU reconfigurations, for instance) that could give rise to two much longer intervals during which, without the proposed technique, all the PDC data would be delayed, thus jeopardizing low-latency application.

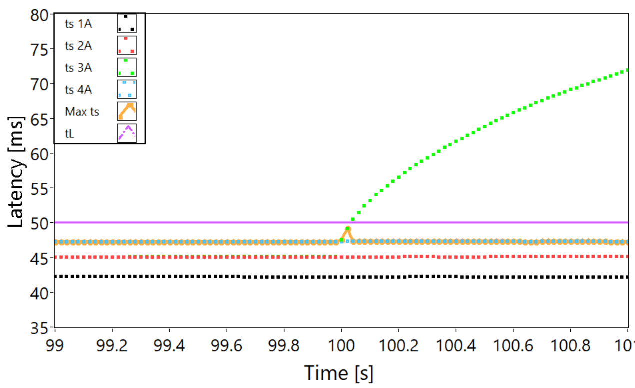

Fig. 7. Trend of the monitor functions relating to the simulated variation of the latency for stream 3A.

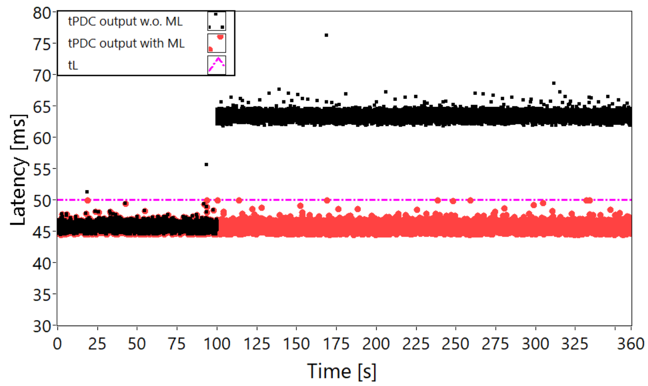

Fig. 8. Output data streams without latency management (tPDC output w.o. ML) and with latency management (tPDC output with ML) in the presence of variation of the latency conditions for stream 3A.

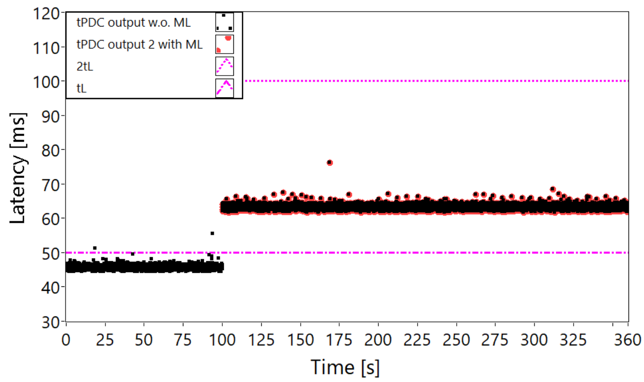

Fig. 9. Output data streams without latency management (tPDC output w.o. ML) and with latency management for the low-priority stream (tPDC output 2 with ML) due to the presence of latency variation.

Fig. 8 shows the data output streams that relate to the three non-delayed PMUs without (tPDC output w.o. ML) and with (tPDC output with ML) the proposed data handling, whose benefits clearly emerge.

Fig. 9 represents the PDC output data of the second stream (tPDC output 2 with ML). In this case, since there are no packet outliers and the latency of PMU 3A remains under the limit $2t_L$, all its data are sent in the same stream.

Similar results hold also when the latency variation is gradual, as shown in Fig. 10 (PMU 3A undergoes a slow latency growth of 0.1 ms/s) for the first, low-latency stream.

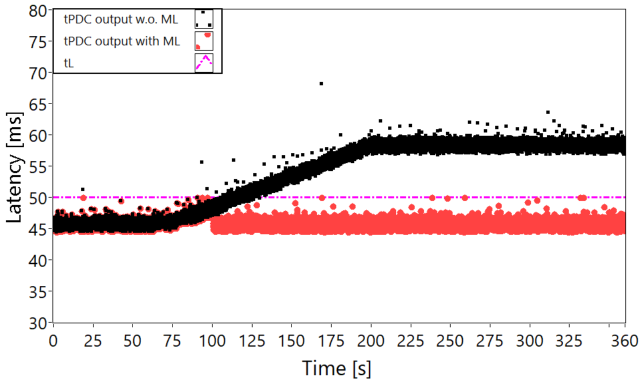

Fig. 10. Output data streams without latency management (tPDC output w.o. ML) with latency management (tPDC output with ML) under gradual latency variation.

5.2 *Heterogeneous PMU scenario*

The PMUs considered in this scenario (see Tables II and III) do not comply with P-class requirements and, therefore a higher time limit, namely $t_L = 150$ ms, has been assumed.

Fig. 11 shows the heterogeneous trend of the latency. PMUs 2B, 3B and 4B are characterised by a steady trend, whereas PMU 1B shows a more variable behavior.

A temporary variation (close to second 40, Fig. 12) of the latency of the PMU 3B stream has been added by means of the network emulator. In particular, a NetEm script has been used with delay of 100 ms, jitter of 40 ms and Pareto distribution.

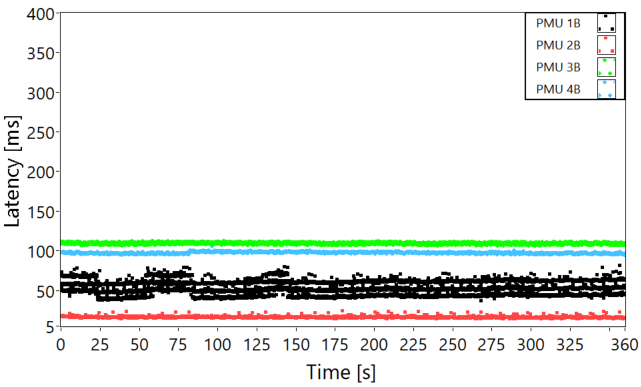

Fig. 11. Data provided by four heterogeneous PMUs.

As a consequence, in Fig. 13, the maximum value of the monitor functions (3) follows that of PMU 3B. When the limit is reached (case 2 of Section 3), stream 3B is separated from the main output stream and this holds until PMU 3B returns below the limit, which happens close to the second 220. During this event, the maximum value of function (3) for the other three PMUs is always below the limit and the main PDC output stream (tPDC output with ML in Fig. 14) includes data from all the PMUs except PMU 3B. It is clear how the proposed policy would allow the target application to operate with three streams instead of losing all the data.

As an example of application, the output, in terms of voltage profile, of a state estimation is shown in Fig. 15 (see [26] for the details on the application). The test is performed by emulation with real PMU prototypes processing signals from an 18-bus radial distribution feeder. Four PMUs are considered as in the heterogenous scenario (placed at nodes 1, 4, 6, 11 of the network considered in [27]). It is assumed that the last PMU undergoes the same latency impairment as described above. If the proposed PDC is used to collect the voltage phasor data, the application can operate, even under latency variation condition, within t_L. State estimation runs using the available data (three PMU measurements) from the high-priority stream and completes the missing information with pseudo-measurements, thus degrading only partially its accuracy, as illustrated by the error bars in the figure (estimated expanded uncertainties). Without the proposed latency management, its outcomes would be practically meaningless, since no PMU data were available, and no estimation could be performed in the required time.

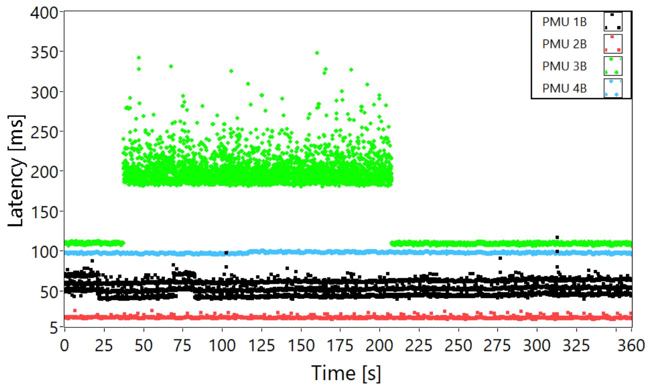

Fig. 12. Data provided by the PMUs in the presence of a temporary variation of the latency conditions for stream 3B (PMU 3B).

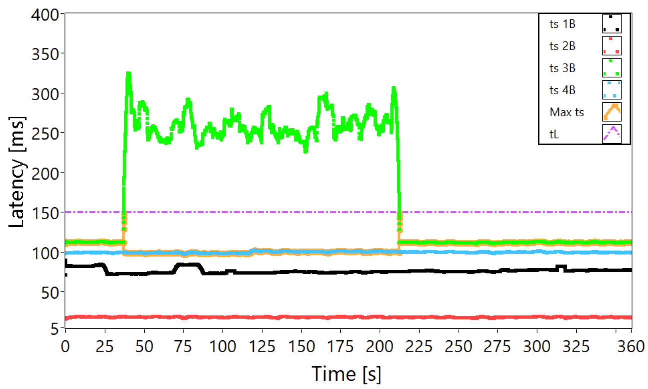

Fig. 13. Trend of the monitor functions relating to the emulated variation of latency for stream 3B.

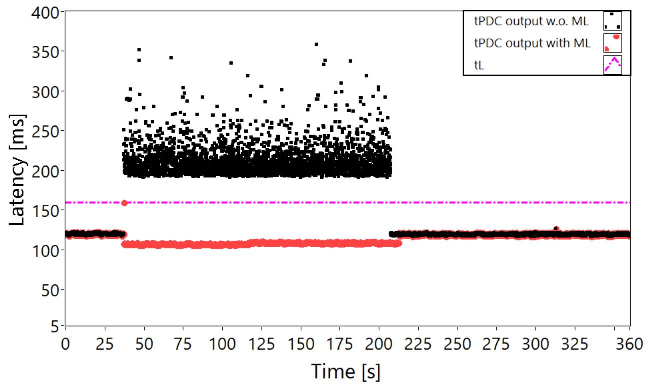

Fig. 14. Output data streams without latency management (tPDC output w.o. ML) and with latency management (tPDC output with ML) in the presence of variation of the latency conditions for stream 3B.

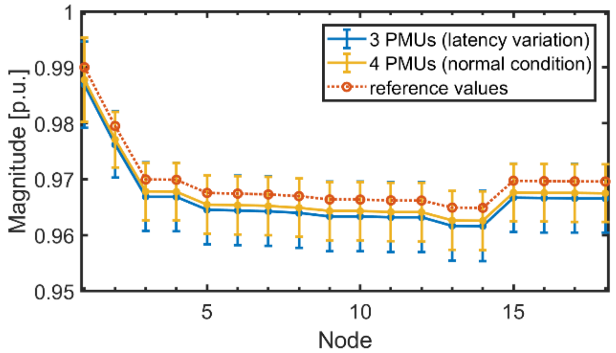

Fig. 15. State estimation application with the proposed PDC: estimated voltage profile with and withour latency variation.

During the event, the data of the second stream, relating to PMU 3B, need to be sent under the newly imposed limit ($2t_L$ in Fig. 16). The individual data received over the second limit are stored and sent, as described above, with low priority for non-real-time applications.

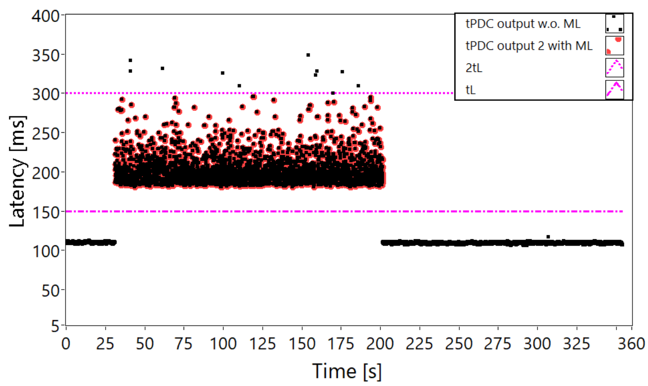

Fig. 16. Output data streams without latency management (tPDC output w.o. ML) and with latency management for the second low-priority stream (tPDC output 2 with ML) due to the presence of latency variation.

As a final test, the PDC prototype computational burden has been analyzed. As clear from previous results, the proposed PDC can manage 4 input streams provided by commercial PMUs at different reporting rates. In Table IV the results of CPU utilization of the PDC process for different reporting rates are summarized.

TABLE IV. CPU UTILIZATION FOR DIFFERENT REPORTING RATES

Reporting Rate [fps]	CPU Utilization [%]	
	Normal condition	Latency variation
10	3.4	3.7
25	7.6	9.4
50	9.4	14.0

Table IV shows the CPU utilization during normal condition and in the presence of the latency impairment. In the first case, all the input data are encapsulated in a single output stream, while in the second case one low-priority stream is separated from the main output stream. The results show how the computational burden increases with the RR in the two conditions. The PDC can also manage 8 different input streams with a CPU utilization below 30 % (with a RR = 50 fps and one stream delayed) and this makes it suitable for the tests. Besides, the processing time obtained with 4000 data packets (with 4 input streams with a RR = 50 fps) is 0.8 ms with a standard deviation of 0.15 ms. These values are indicative, and can obviously vary if further functionalities and computations are implemented in the PDC, but are useful to give an idea of the PDC operation.

6. CONCLUSIONS

An innovative active PDC has been presented, with advanced functionalities that allow adaptively managing the latency of several input PMU streams by acting also on output data aggregation. Several simulations and experimental tests have been performed to validate the approach. The results show how endowing the PDC with synchronization and latency monitoring capabilities allows safeguarding the requirements of the real-time applications based on modern wide area monitoring systems for power networks.

The PDC is thus no more a simple routing element, but becomes a real measurement device, able to fit the time needs of the monitoring architecture.

Such approach would also make it possible to extend the concept of latency requirements from the compliance of the individual PMU, as indicated by the synchrophasor standard IEEE C37.118.1, to the communication layer or even to the whole architecture, if PDCs establish peer to peer dialogs and can thus coordinate with each other.

ACKNOWLEDGMENT

The research activities described in this paper have been partially funded within the R&D project "Cagliari2020" (partially funded by the Italian University and Research Ministry, grant# MIUR_PON04a2_00381).

REFERENCES

[1] "IEEE Standard for Synchrophasor Measurements for Power Systems", IEEE Std. C37.118.1-2011, pp. 1–61, Dec. 2011. IEEE Std. C37.118-2005 Revis.

[2] "IEEE Standard for Synchrophasor Data Transfer for Power Systems", IEEE Std C37.118.2-2011, pp. 1–53, Dec. 2011. IEEE Std C37.118-2005 Revis.

[3] P. Castello, P. Ferrari, P. A. Pegoraro, S. Rinaldi, "Hardware for PMU and PMU Integration", in *Phasor Measurement Units and Wide Area Monitoring Systems*, 1st ed., A. Monti, C. Muscas, and F. Ponci, Eds. Elsevier, Academic Press, 2016, Ch. 5, pp. 63–86.

[4] "IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring", IEEE Std C37.244-2013, pp. 1–65, May 2013.

[5] "Reliability Guideline, PMU Placement and Installation", Dec 2016, Available online, http://www.nerc.com/pa/RAPA/rg/ReliabilityGuidelines/Reliability%20Guideline%20-%20PMU%20Placement.pdf

[6] A. Angioni et al. "A distributed automation architecture for distribution networks, from design to implementation", *Sustainable Energy, Grids and Networks* (2017), DOI: 10.1016/j.segan.2017.04.001.

[7] M. Pignati *et al.*, "Real-time state estimation of the EPFL-campus medium-voltage grid by using PMUs", in *Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power Energy Society*, 2015, pp. 1–5.

[8] A. von Meier, D. Culler, A. McEachern, and R. Arghandeh, "Micro-synchrophasors for distribution systems", in *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*, 2014, pp. 1–5.

[9] S. Rinaldi, et al. "Distributed monitoring system for voltage dip classification over distribution grid", *Sustainable Energy, Grids and Networks*, 6 (2016) 70-80.

[10] C. H. Lo and N. Ansari, "The Progressive Smart Grid System from Both Power and Communications Aspects", in *IEEE Communications Surveys & Tutorials*, vol. 14, no. 3, pp. 799-821, Third Quarter 2012.

[11] Y. Wu, L. Nordström, "Adaptive data link configuration for WAMC applications using a Stateful Data Delivery Service platform", *Sustainable Energy, Grids and Networks,* 6 (2016) 166-175.

[12] A. Meloni, P. A. Pegoraro, L. Atzori, A. Benigni and S. Sulis, "Cloud-based IoT solution for state estimation in smart grids: Exploiting virtualization and edge-intelligence technologies", *Comput. Netw.*, vol. 130, pp. 156-165, 2018.

[13] M. S. Thomas, N. Senroy, and A. S. Rana, "Analysis of time delay in a wide-area communication network", in *2014 6th IEEE Power India International Conference (PIICON)*, 2014, pp. 1–6.

[14] M. G. Adamiak, M. Kanabar, J. Rodriquez, and M. D. Zadeh, "Design and implementation of a synchrophasor data concentrator", in *2011 IEEE PES Conference on Innovative Smart Grid Technologies - Middle East (ISGT Middle East)*, 2011, pp. 1–5.

[15] A. Derviskadic; P. Romano; M. Pignati; M. Paolone, "Architecture and Experimental Validation of a Low-Latency Phasor Data Concentrator", in *IEEE Trans. Smart Grid*, vol.PP, no.99, 2016.

[16] K. Zhu, S. Rahimi, L. Nordström, B. Zhang, "Design phasor data concentrator as adaptive delay buffer for wide-area damping control", *Electric Power Systems Research*, vol. 127, 2015, pp. 22-31.

[17] R. Pourramezan; Y. Seyedi; H. Karimi; G. Zhu; M. Mont-Briant, "Design of an Advanced Phasor Data Concentrator for Monitoring of Distributed Energy Resources in Smart Microgrids", in *IEEE Trans. Ind. Informat.*, vol.PP, no.99, 2016.

[18] P. Castello, C. Muscas, P. A. Pegoraro, and S. Sulis, "Adaptive Management of Synchrophasor Latency for an Active Phasor Data Concentrator", 2017 *IEEE International Instrumentation and Measurement Technology Conference Proceedings*, Torino, 2017, pp. 1-5.

[19] P. Castello, C. Muscas, P. A. Pegoraro, and S. Sulis, "Automated test system to assess reporting latency in PMUs", in 2016 *IEEE International Instrumentation and Measurement Technology Conference Proceedings*, Taipei, 2016, pp. 1–6.

[20] P. Castello, J. Liu, C. Muscas, P. A. Pegoraro, F. Ponci, and A. Monti, "A Fast and Accurate PMU Algorithm for 'P+M' Class Measurement of Synchrophasor and Frequency", *IEEE Trans. Instrum. Meas.*, vol. 63, no. 12, pp. 2837–2845, Dec. 2014.

[21] E. Price, "Practical considerations for implementing wide area monitoring, protection and control", in *Protective Relay Engineers, 2006. 59th Annual Conference for*, 2006.

[22] G. Celli, P. A. Pegoraro, F. Pilo, G. Pisano and S. Sulis, "DMS Cyber-Physical Simulation for Assessing the Impact of State Estimation and Communication Media in Smart Grid Operation," in *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2436-2446, Sept. 2014.

[23] "EPFL Smart Grid - The Vision | EPFL Smart Grid". [Online]. Available: http://smartgrid.epfl.ch/ [Accessed: June-2017].

[24] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in HAN, NAN and WAN", *Comput. Netw.*, vol. 67, pp. 74–88, Jul. 2014.

[25] A. G. Phadke and J. S. Thorp, "Communication needs for Wide Area Measurement applications", 2010 *5th International Conference on Critical Infrastructure (CRIS)*, Beijing, 2010, pp. 1-7.

[26] P. A. Pegoraro, A. Meloni, L. Atzori, P. Castello and S. Sulis, "PMU-Based Distribution System State Estimation with Adaptive Accuracy Exploiting Local Decision Metrics and IoT Paradigm," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 4, pp. 704-714, April 2017.

[27] A. R. Ahmadi, T. C. Green, "Optimal power flow for autonomous regional active network management system", Proc. IEEE Power Energy Soc. General Meeting, July 2009, pp. 1-7.