A Tool to Characterize Delays and Packet Losses in Power Systems with Synchrophasor Data

Christoph Lackner, Student Member, IEEE, Felipe Wîlches-Bernal, Member, IEEE, Brian J. Pierre, Member, IEEE, and David A. Schoenwald, Senior Member, IEEE

Abstract—This paper describes the implementation of a tool to estimate latencies and data dropouts in communication networks transferring synchrophasor data defined by the IEEE C37.118 standard. The tool time tags synchrophasor packets at the time they are received, according to a GPS clock, and with this information is able to determine the time those packets took to reach the tool. The tool is able to connect simultaneously to multiple PMUs sending packets at different reporting rates with different transport protocols such as UDP or TCP. The tool is capable of redistributing every packet it receives to a different device, while recording the exact time this information is re-sent into the network. The results of measuring delays from a PMU using this tool are presented and compared with those of a conventional network analyzer. The results show that the tool presented in this work measures delays more accurately and precisely than the conventional network analyzer.

Index Terms—Synchrophasors, PMU, GPS, Delay, Latency, Communication Networks

I. INTRODUCTION

Synchrophasor technology, originating in the late 1980s, allows high resolution monitoring of power systems spanning large geographical areas [1]. Phasor measurement units (PMUs), the building block of this technology, continuously measure relevant power system quantities. PMUs compute phasors out of these measurements and time tag each measurement according to a common and highly accurate time reference. The IEEE C37.118 standard defines the requirements for the transmission of synchrophasor data [2], [3] and specifies the accuracy needed in the Coordinated Universal Time (UTC) time reference.

Synchrophasor data is prevalent due to the extensive installation of PMUs in power grids all around the world as in the US [4] and China [5]. It is being used for many applications from complementing SCADA data in power system state estimators [6], [7], to disturbance monitoring [8], to wide-area situational awareness [9], to wide-area control [10]–[13]. Because PMU data is time tagged according to a common reference it can be synchronized even when stemming from distant sources. PMUs are the backbone of the future smartgrid at the transmission level of bulk power systems [4]. It is important to note that the PMU data is only available for use after a certain delay. This delay has different sources: the PMU itself, which is related to the time the device takes to process measurements and compute the phasor, and the communications medium over which the synchrophasor data is transmitted [14], [15]. In some applications such as wide-area control calculating and understanding the nature of these delays is extremely important as longer and random delays may impact the performance of a control system negatively if not addressed properly [14]–[24]. Some work has successfully analyzed how to circumvent some of the problems associated with delays and packet losses in PMU data for real-time control [25], [26]. Note that the tool presented in this work has been used in the network characterization and design of the wide-area controller in [10]–[12], [27]. It is important to highlight that accurately characterizing delays in a control system and the source of these latencies is invaluable information. If design engineers have precise knowledge of where each delay is coming from they can propose architectures to try to reduce those delays. In addition, real-time control systems can be designed in a way to counter or mitigate the destabilizing effect of delays if they are well characterized. Additionally, some work has been done in modeling the delays in communication networks [28], and it has been shown in simulations that knowledge of the characteristics of delays can be used to identify cyber security attacks in power system communication networks [29]. Understanding the underlying structure of the delays and distinguishing between delays caused by the network and those caused by the PMU can improve the modeling of these delays. This distinction can be used to model delays in wide area monitoring systems with different communication methods before installing expensive communication equipment and PMUs.

A considerable amount of work has been done on how to measure delays in communication networks [30]. A number of tools, such as Wireshark (WS), are capable of measuring the time of arrival of packets and determine delays. However these tools were developed to characterize general communication networks, such as the Internet. Synchrophasor data already has a very accurate Global Positioning System (GPS) time stamp (TS), which can be leveraged to determine delays with a much higher accuracy. A more accurate delay measurement can be used in a number of applications mentioned above and can result in a more detailed delay model. Tools such as the Synchrophasor Stream Splitter (SPSSP) or the PMU Connection Tester [31], have been developed to analyze Synchrophasor...
data in communication networks. However, these tools do not account for delays and can add a significant delay to the data. These types of effects are analyzed and shown in this paper.

This paper details the design considerations and implementation of a tool to accurately measure the delays of synchrophasor data in IP based communication networks. The tool, named Network Characterizer (NC), is able to interpret data as specified in the C37.118 format. In addition, the tool is connected to a GPS clock in UTC so it is able to time tag every packet it receives. The contributions of this paper include:

- Presenting the NC tool and showing that it measures delays more accurately than an alternative network analyzer. This is shown for data from a PMU using either the transmission control protocol (TCP) or user datagram protocol (UDP).
- Presenting a NC tool that is able to align PMU data packets by their GPS TS and identify any missing or lost packets.
- Describing in detail the design considerations to design a tool to accurately measure delays of synchrophasor data in real-time. It also provides detailed descriptions of how to incorporate PMU packet arrival times in the design of a network characterizer tool for accurately measuring delays.
- Introducing additional functionality to the NC tool that allows it to repeat (or mirror) every packet it receives and record the time the packet was sent into the network. This action is performed without adding any major delay to the mirrored data. Note that having an accurate time tag of the moment a packet is sent into a network is important to measure the communications or network delay of synchrophasor packets without including the processing or phasor computation delay.
- Presenting measurements of delays of actual PMU data for both the UDP and TCP protocols at different reporting rates and performing statistical analysis of them.
- Presenting measurements of synchrophasor data delays and packet losses of a Local Area Network (LAN) and comparing it with those coming from the public Internet.

The remainder of this paper is organized as follows. Section II briefly introduces the C37.118 standard that defines synchrophasor data transfer. Sections III and IV present the description and implementation of the network characterizer, respectively. Section V shows some measurements of the delays obtained with the NC for different transport protocols. Finally, Section VI outlines the conclusions of this work and elaborates future directions of research.

II. THE IEEE C37.118.2 STANDARD FOR SYNCHROPHASOR DATA TRANSFER

The IEEE C37.118.2 standard defines the requirements for measurements and data transfer of synchronized phasor measurements in power systems [2]. It is used by most PMUs to transmit synchrophasor data from substations to control centers or wide area actuators in the electrical network.

A. Relevant Definitions and Specifications in the Standard

The C37.118.2 standard distinguishes between multiple types of frames for data transfer and coordination. Three of those frames are relevant for the NC:

1) Data Frame: Data frames contain a GPS TS and the values of analog and digital measurements taken at that time, as well as status information of the PMU. Data frames are sent at a fixed data rate, and one dataframe has to be sent at the top of every second [3].

2) Configuration Frame: Configuration frames contain the data rate of the PMU and the timebase divisor TIME_BASE, which is used to compute the fractional component of the TS in all frames [3]. If the spontaneous data transmission method and UDP are used, the PMU sends a configuration frame once every minute, otherwise they have to be requested through a command frame.

3) Command Frame: Command frames are sent from the NC to the PMU to request configuration frames, a start of data transfer or an end of data transfer. They are only used if TCP and the non-spontaneous data transmission are used.

B. Transport Protocols

While the C37.118.2 standard defines the content of the data packets, it allows any transport protocol for performing the data transfer. These transport protocols allow the standard to define frames independently of what data transmission medium is used. They can include error checking, network congestion control and other low level functionality that is not provided by the standard. The most common transport protocols in IP-based communication networks are TCP and UDP. Synchrophasor data transfer can occur through either of them.

UDP is a connectionless and packet-based type of protocol where the speed of data transmission is high. In most real-time synchrophasor applications UDP is used because it provides low communication delays. However, if a packet is lost in the network UDP does not re-send it [32]. Note that with UDP it is possible that packets are received in a different order than they are sent.

TCP is a connection-oriented and stream-based protocol that guarantees the reception of packets by the receiver. This is achieved through acknowledgments and data re-sends. This feature increases the communication overhead, which increases the communication delays. Note that with TCP there are no data dropouts and the information is always available in the same order it is sent [33].

III. DESCRIPTION OF THE NETWORK CHARACTERIZER

The NC presented in this paper can receive and parse C37.118.2 packets simultaneously from multiple PMUs with different data rates using either the TCP or UDP protocols. Every time the NC detects a packet of information has arrived it timestamps it according to a GPS clock in UTC. This new time tag is defined as the time of arrival (TOA). The NC constantly logs the following data for every PMU it analyzes:

- the TS that comes within a particular PMU frame,
- the TOA,
• the Data Quality Flag (DQF), which indicates if a Phasor Data Concentrator (PDC) added a delay to the frame [34], and,
• the Time Quality Flag (TQF), which indicates if a PMU has lost GPS synchronization.
In addition to this logging functionality, the NC is able to redistribute any PMU data stream it receives as many times as necessary. That is, any incoming data stream can be redirected as needed to different devices. This feature is similar to the SPSSP. Whenever a PMU data stream is being redirected by the NC, the NC in addition to recording the 4 fields mentioned above, also records the time the packet was sent into the network. This time is named time of re-send (TRS).

Previous works on PMU delays have defined a separation of the total delay into a communication latency $t_{com}$ and a PMU or measurement delay $t_{meas}$ [15]. The repeater or splitting capability of the NC can be used to distinguish between these two delays.

The NC tool was implemented in the NI-developed software-design platform and development environment LabVIEW [35]. The tool is composed of:

• A synchronous part which is run in a NI Real-Time PXI (RT-PXI). This part timestamps, collects and decodes the packets as described in Section II.
• An asynchronous part which is run on a computer connected to the RT-PXI. This part is in charge of aligning the data by PMU TS and recording it. In addition, this part has a graphical interface to display delays, TS interpacket latencies (IPLs), and TOA IPLs.

The separation into an asynchronous and a synchronous part allows the NC to receive packets and timestamp them on a Real-Time (RT) operating system (OS). The RT OS guarantees that no delays are added to the packet when it comes out of the network card into the NC software. The use of the RT-PXI allows a GPS clock to be connected to the NC. This increases the accuracy since then the PMU TS as well as the TS applied after the packets are received have to be synchronized within 25 $\mu$s to comply with the IEEE C37.118.1 standard [2]. Fig. 1 shows the setup of the NC with the RT-PXI and the computer running the synchronous and asynchronous parts, respectively.

IV. IMPLEMENTATION OF THE NETWORK CHARACTERIZER
This Section presents the details in the implementation of each of the parts of the NC.

A. The RT or Synchronous Part
The synchronous part of this tool is responsible for:

• Initializing the network interface to listen for packets,
• sending command frames if TCP is used,
• receiving the packets from the network interface,
• time stamping each packet on arrival with a GPS time in UTC,
• decoding the configuration packets to obtain the data rate and timebase,
• decoding the IEEE C37.118.2 packets to obtain the TS,
• repeating the packet back to the network if desired, and time stamping the packet with a TRS.

The remainder of this section describes these tasks in more detail.

1) Initialization: When the NC is started the RT-PXI reads a configuration file to obtain the data needed to initialize network connection sockets. This file consists of two sections:
   (i) Network connection information, and
   (ii) PMU setup information

Table I describes the connection information portion of the initialization file. When UDP is used, it can be used in multicast or unicast. Table II describes the PMU setup information in the initialization file. The NC checks the PMU ID and the IP address against the information contained in the frames to ensure the frames belong to the correct PMU.

<table>
<thead>
<tr>
<th>Table I</th>
<th>Connection section in the .ini file</th>
</tr>
</thead>
<tbody>
<tr>
<td>localIPaddress</td>
<td>The IP address of the RT-PXI on which the PMU packets are received</td>
</tr>
<tr>
<td>multicastAddress</td>
<td>The multicast group address on which PMU packets are received if multicast is used</td>
</tr>
<tr>
<td>remotePort</td>
<td>The port to which the PMU packets are being sent</td>
</tr>
<tr>
<td>device*</td>
<td>Entry for each PMU that is connected to this UDP port e.g. PMU01</td>
</tr>
<tr>
<td>repeaterPort*</td>
<td>The port to which the PMU packets are repeated to</td>
</tr>
<tr>
<td>repeaterIPaddress*</td>
<td>The IP address the PMU packets are repeated to</td>
</tr>
<tr>
<td>TCP Section</td>
<td>Description</td>
</tr>
<tr>
<td>localIPaddress</td>
<td>The IP address of the RT-PXI on which the PMU packets are received</td>
</tr>
<tr>
<td>remotePort</td>
<td>The port to which the PMU command frames are being sent</td>
</tr>
<tr>
<td>device01</td>
<td>The PMU that is connected to this TCP connection (only one PMU can be connected to a single TCP connection)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table II</th>
<th>PMU information section in the .ini file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>remoteIPaddress</td>
<td>IP address of the PMU</td>
</tr>
<tr>
<td>idcode</td>
<td>PMU ID*</td>
</tr>
<tr>
<td>pmuName</td>
<td>PMU Name used in analysis</td>
</tr>
<tr>
<td>datum*</td>
<td>Name of data channel n*</td>
</tr>
<tr>
<td>datum*type</td>
<td>Type of data channel n*</td>
</tr>
<tr>
<td>scaling</td>
<td>Scaling of data*</td>
</tr>
</tbody>
</table>
* as defined in [3]
2) Configuration initialization: Since the timebase is contained in the configuration packet, the NC needs to receive a configuration packet for a particular PMU before being able to process any data packets from it. When started the NC receives packets and timestamps all arriving packets, however all data packets are discarded until a configuration packet has arrived. When a configuration packet is received, typically at the top of the minute, the NC determines the timebase and the data rate of the PMU.

For some PMUs and PDCs the configuration packets do not arrive at the top of every minute. In order to synchronize the analysis of each PMU the NC waits until all PMUs have sent a configuration frame before analyzing any packets. To avoid waiting indefinitely if a PMU is disconnected or not sending configuration frames, the NC only waits for a maximum of 120 s before removing any PMUs from which it has not received a configuration frame.

Fig. 2a shows an example where 4 PMUs using UDP are being analyzed. PMU A, B and C send a configuration packet at the top of every minute, however PMU D sends the configuration packet 15 s later. The NC is started at $t = 5$ s but does not start analyzing the packets until $t = 60$ s since it waits until all 4 configuration packets have been received.

Fig. 2b shows the same scenario when PMU A is disconnected from the network. At $t = 125$ s the NC has been running for 120 s, and no configuration packet from PMU A was received; Therefore the NC removes PMU A and starts analyzing the remaining 3 PMUs.

Fig. 2c shows the same scenario but in this case a configuration packet from PMU B was lost in the network. Because the NC waits until $t = 125$ s before starting the analysis, the second configuration packet from PMU B is received and the analysis performed by the NC includes PMU B.

Data frames are decoded to obtain TS, TQF and DQF. This information is then sent to the synchronous part through the secondary network interface where the information is further processed as described in Section IV-B.

4) Repeater functionality: If the repeater functionality is active the synchronous loop also repeats the packet to a user determined IP address as described in Section IV-A1. When the packet is sent, a GPS timestamp (TRS) is taken and sent to the asynchronous part as described in Section IV-A3.

B. The Asynchronous Part

The asynchronous part of the NC consists of a graphical interface and a logging mechanism. Fig. 3 shows the interface which contains 3 plots. These plots continuously show the latency, IPLs of TOAs and IPLs of TSs, received in the last 2 minutes.

The graphical interface offers the option to save the configuration packets’ TSs and TOAs in a separate file if needed for further analysis. In order to allow analysis of the collected data, the NC records TS, TOA, DQF and TQF in a comma separated value file. The NC also identifies packets lost in the network at this stage and records them as missing. The recorded data is aligned by expected TS and combined in a single file for each data rate. If multiple data rates are present the NC writes one file for every data rate. The data is processed and aligned one second at a time, independent of the data rate. Hence, the data file is then written every second. In order to avoid large file sizes, a new file is generated every 10 minutes. The desired file size can easily be adjusted if a larger time frame per file is necessary.

The remainder of this section describes the processing of data packets for a single data rate. If multiple PMUs with different data rates are characterized this process is done for each data rate independently.

1) Data Alignment: In order to identify missing packets, packets received with an extremely high delay, and packets received out of order, it is necessary to know when packets
are expected to arrive. Because the IEEE C37.118.2 standard requires PMUs to timestamp the measurements at the top of every second and evenly distribute within a second, it is possible to compute the expected TS given the data rate $d$ of a PMU by,

$$t_i = t_{\text{start}} + \frac{i}{d} \quad \forall i = 0, 1, \ldots$$

(1)

where $t_{\text{start}}$ is the first TS which is analyzed and $t_i$ is the expected TS of the $i$th packet. Section IV-B3 describes how $t_{\text{start}}$ is obtained in more detail. Due to numerical inaccuracies it is possible for the RT-PXI to decode some TS slightly different from the expected TS. In order to avoid removing these packets and classifying them as missing, the tool only requires the expected TS and the actual TS to be within $\frac{1}{2dT}$ of each other, where $d$ is the data rate. Fig. 4 shows a number of expected TS and actual TS as well as the interval in which a TS is accepted as matching the expected TS.

![Fig. 4. TS vs. expected TS with tolerance and numerical inaccuracy.](image)

Because some data frames have large delays from the time the packet is timestamped by the PMU until it is analyzed by the NC, the NC waits until it has collected at least 6 s of data from all PMUs before attempting to align the packets according to the expected TS. This results in packets being aligned correctly even if they have network delays on the order of seconds. However, if a PMU has a delayed packet that took more than 6 s longer than most other packets, the packet is identified as lost and discarded. This is necessary to allow the NC to identify when a packet is lost in the communication network and allows an analysis of the network reliability. By recording all expected TS even if no corresponding packet has arrived from the PMU, it is possible to identify packets lost due to network congestion.

In the following examples the frame labeled $A_i$ corresponds to the $i$th data frame sent by PMU A whose TS is computed by (1). All PMUs use representative data rate of $d = 3$ frames per second (fps). Fig. 5a shows an example when frame $A_2$ is delayed by 6.2 s. At $t = 7.33$ s the NC received frame $A_{21}$ and has 20 frames or 6 s of data. As previously described, the NC aligns frames $A_1$ to $A_3$ even though frame $A_2$ is missing. This frame is identified as missing and once it arrives at $t = 7.5$ s, it is discarded. Fig. 5b shows an example with the same frame being delayed by about 5.2 s. In this case the NC only has 17 frames of data once frame $A_2$ arrives and continues to collect data until frame $A_{20}$ arrives. At $t = 7$ s frames $A_1$ to $A_3$ are aligned and no frames are identified as missing.

In order to avoid a memory leak, if a PMU is disconnected, the NC also aligns data once it has received more than 20 s of data from at least one PMU. Fig. 5c shows an example where PMU B is disconnected at $t = 4$ s. This results in frames $B_{11}$ and higher missing. At $t = 4$ s the NC has received 3 s of data from PMU A and PMU B. Since there are no more frames coming from PMU B the NC will never have 6 s of data. However, at $t = 21$ s the NC has 20 s of data from PMU A and starts to align frames $A_1$ to $A_3$ and $B_1$ to $B_3$. This will be repeated with frames $A_4, B_4$ to $A_6, B_6$ and frames $A_7, B_7$ to $A_9, B_9$ at $t = 22$ s and $t = 23$ s, respectively. From this point on all frames from PMU B are identified as missing and the NC only aligns frames from PMU A.

Both settings, the minimum time of available data as well as the maximum, can easily be adjusted. This allows the NC to characterize networks with significantly different behaviors.

2) Removal of stale data: In order to account for cases where individual packets are delayed by more than 6 s the NC records all packets that can not be aligned with an expected TS in a separate file. When aligning data it is also necessary to remove data with a TS that can not be aligned. That is, any data with a TS earlier than the last aligned TS is removed after aligning one second of data. For instance, when aligning frames with $4 \leq \text{TS} < 5$ s any frames with $\text{TS} < 5$ s are removed.

3) Identification of the initial TS: As described in Section IV-B1 an initial TS is necessary to determine the expected TS. Since a packet has to be timestamped at the top of every second it is convenient to start with a full second [3]. However, in order to ensure all PMUs have started sending data when the NC starts recording, it is necessary to identify a TS which is present in the data for all PMUs. Once all PMUs have sent at least 1 data frame and a configuration frame, as described in Section IV-A2, the TSs are combined and the highest (most...
recent) TS is selected. By adding 1 s to the maximum TS and rounding down to the closest full second, it is guaranteed that all PMUs have already taken the measurement with this TS and the frame has arrived at the NC or is in the network. If the NC receives data for a single PMU for more than 20 s before receiving any data for the other PMUs, an issue arises since the NC starts to align data immediately, as described in Section IV-B1. In this case data for the other PMUs is identified as missing. To avoid this, the maximum data size at which the NC starts aligning the data as described in Section IV-B1 is increased to 140 s during the alignment of the initial second of data. Assuming all PMUs are connected this is sufficient to guarantee all PMUs have sent more than 6 s of data with a TS greater than the initial TS.

To show this, Fig. 6 considers the worst case scenario. Three PMUs are being analyzed with $d = 10$ fps. The configuration frame for PMU A and B is received at $t = 0$ s when the NC is started. The NC collects data from these PMUs starting with TS = 0, but the data is not analyzed until the configuration packet from PMU C is received which is 120 s later. Fig. 6 shows the case when the configuration frame for PMU C is received at $t = 120$ s and a maximum dataset of 20 s is used as described in Section IV-B1. When the first data packet from PMU C is received with TS = 120 s the data is analyzed. The initial TS in this case is $t_{start} = 121$ s. Since the NC already has 120 s of data from PMUs A and B, it starts aligning data for $121 \leq TS < 122$ s. However, this data has not been received by the NC and results in missing data indications for all PMUs. By requiring at least 140 s of data from a single PMU during the initial alignment it is guaranteed that the NC waits until it has received 6 s of data from all PMUs, or at least 20 s of data from one PMU, with TS > $t_{start}$. These cases are seen in Figs. 6b and 6c, respectively. This is the worst case scenario since the last configuration packet has to arrive within 120 s of the first one as described in Section IV-A2, otherwise the PMU is removed from the analysis.

V. MEASUREMENT OF LATENCIES WITH THE NC

The NC was used to characterize the latencies of a PMU in a simple experimental network. The results presented in Sections V-A and V-B correspond to the PMU using either TCP or UDP respectively. Section V-C presents a comparison of the effect of the SPSSP and the repeater functionality of the NC.

Fig. 7 shows the network setup used in these tests. For the results in Section V-A the PMU sent UDP multicast packets which were analyzed by the NC on the RT-PXI and WS on a desktop computer simultaneously. For the results in Section V-B the PMU sent two separate TCP streams which where analyzed by WS and the NC simultaneously. For the results in Section V-C the PMU was sent UDP multicast packets, a NC with repeater functionality and a desktop computer with the SPSSP were used to repeat the packets to another NC. The data recorded by this last NC was used to compare the delays. Because in all three cases the network only consisted of a single switch which was used by synchrophasor data traffic only, no packets were lost.

![Network Setup](image)

(a) used for WS to NC comparison.

(b) used for SPSSP and repeater functionality comparison.

Fig. 7. Experimental network setup.
Since the networks in both configurations in Fig. 7 are fully located within a laboratory they do not involve large delays or congestions. The results in Sections V-A to V-C do not contain any missing packets due to this lack of network congestions. Section V-D contains a comparison of the delays when a LAN is used vs. the public Internet. Fig 18 shows the network used for these tests.

A. UDP Protocol Results

This section describes the delays measured when using UDP. The results were obtained by using UDP multicast with PMU data rates of 30 and 60 fps. The delays are calculated by,

\[ T_{DEL} = TOA - TS \]  \hspace{1cm} (2)

where the TS is taken from the PMU packet and the TOA is measured by the NC and WS respectively.

Fig. 8 shows the delays measured when the PMU was set to a frame rate of 30 fps. Fig. 8a shows that the delays measured by WS increase over time due to a drift in the clock used to determine the TOA. The desktop computer running WS was set to synchronize to a Network Time Protocol (NTP) server every 10 minutes which caused the large jumps in the delays. The drift of the clock was estimated to be 3.67 µs. This estimation was used to remove this drifting effect in Figs. 8c and 8d. Figs. 8c and 8d indicate that the delays measured by the NC vary less than the those measured by WS, even after removing the clock drift.

It should be highlighted that this variation in the expected value of the delay will influence the distributions in Figs. 9a and 9b. In order to avoid this effect on the distributions, Figs. 9c and 9d show the distribution of the delays over a reduced (15 s) interval. During this short period of time the seasonality of the process can be neglected. These results show that the delays measured with the NC have a lower variation than those measured with WS, even after the clock drift is removed. WS runs on a desktop computer and OS can introduce random delays in the packets being received. Because the NC runs on a RT-OS, the OS will not introduce an additional delay. The standard deviations in Table III confirm these results.

Fig. 10 shows the delays measured when the PMU was set to a frame rate of 60 fps. Similarly to the 30 fps case the delays measured by WS increase over time. The drift estimated using the 60 fps data was very close to the drift estimated using the 30 fps data since this phenomenon is not affected by the data rate of the PMU. Figs. 10c and 10d show the delays after the drift is removed. These results show that the

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
 & mean (ms) & median (ms) & std dev (ms) \\
\hline
NC & 44.34 & 44.35 & 0.1557 \\
WS & 44.56 & 44.52 & 0.4893 \\
WS w/o drift & 44.38 & 44.34 & 0.4383 \\
\hline
\end{tabular}
\caption{Statistical Results for NC and WS using UDP at 30 fps over 15 s.}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
 & mean (ms) & median (ms) & std dev (ms) \\
\hline
NC & 27.51 & 29.22 & 2.066 \\
WS & 27.50 & 28.93 & 2.160 \\
WS w/o drift & 27.37 & 28.71 & 2.176 \\
\hline
\end{tabular}
\caption{Statistical Results for NC and WS using UDP at 60 fps over 15 s.}
\end{table}
measurements taken by the NC have less variations than those taken by WS. Figs. 11a and 11d show the PDF and CDF of the measurements, respectively. The results of the mean delay show that synchrophasor delays are lower at higher PMU data rates. Since the same network was used with both data rates, this indicates the internal PMU delay, corresponding to the measurement and processing of the data, is data rate dependent. The mean and median values in Table IV confirm this result. The standard deviations in Table IV also show that synchrophasor delays are lower at higher PMU data rates.

B. TCP Protocol Results

This section describes the delays measured when using TCP. The results were obtained by two TCP streams originating from the same PMU which were analyzed by a desktop computer running WS and the NC simultaneously and independently. The PMU data rate was set to 30 fps and 60 fps.

Fig. 12 shows the delays measured by both WS and the NC when using a data rate of 30 fps. Similar to the UDP, a drift in the computer’s clock causes the delays measured by WS to increase over time. However, due to the fast variations of the delay it is difficult to estimate the drift in this case. Fig. 12b shows the delays right after the synchronization.

Fig. 13 shows the statistical distribution of the delays measured by WS and the NC. Table V shows a summary. From this it can be seen that the delays measured by WS vary more than the delays measured by the NC. The distribution of the delays measured with WS in Fig. 13a has a flat tail towards higher delays that is not shown in the figure.

<table>
<thead>
<tr>
<th>TABLE V</th>
<th>STATISTICAL RESULTS FOR NC AND WS USING TCP AT 30 fps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean (ms)</td>
</tr>
<tr>
<td>NC</td>
<td>55.96</td>
</tr>
<tr>
<td>WS</td>
<td>66.46</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE VI</th>
<th>STATISTICAL RESULTS FOR NC AND WS USING TCP AT 60 fps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean (ms)</td>
</tr>
<tr>
<td>NC</td>
<td>39.54</td>
</tr>
<tr>
<td>WS</td>
<td>50.13</td>
</tr>
</tbody>
</table>

Fig. 14 shows the delays measured when using a data rate of 60 fps. Similar to the UDP case, the results show that the delays are slightly lower when using a higher frame rate. Fig 15 shows the statistical distribution of the delays measured by both WS and the NC. Based on the PDF in Fig. 15a it can be seen that the delays measured by the NC show less variation.
than those measured by WS. Table VI shows the mean and median values as well as the standard deviation measured over 30 minutes. Since the delays measured by WS drift over a long time, the mean and median delay are significantly higher than those measured by the NC. This same result holds for the standard deviation.

![Fig. 14. Delays measured by NC and WS when using TCP at 60 fps.](image)

![Fig. 15. Distribution of the delays measured by NC and WS when using TCP at 60 fps.](image)

**C. Results of Splitter functionality**

The results in this section contain a comparison of the delays when the packets are sent through the SPSSP and the NC with repeater functionality.

Figs. 16a and 16b show the delays measured by the second NC. Figs. 16c and 16d show the delay added by the NC repeater functionality and the SPSSP. This delay is obtained by

\[ T_D = T_{\text{DEL,NC}} - T_{\text{DEL,PMU}} \]  

\[ T_{\text{DEL,SPSSP}} = T_{\text{DEL,SPSSP}} - T_{\text{DEL,PMU}} \]

where \( T_{\text{DEL,PMU}} \) is the delay measured when the packets come directly from the PMU and \( T_{\text{DEL,NC}} \) and \( T_{\text{DEL,SPSSP}} \) are the delays measured when the packets are replicated by the NC repeater functionality and the SPSSP, respectively.

Table VII and Fig. 17 show the statistical distribution of these delays. From this it can be seen that the NC repeater functionality adds less delay than the SPSSP. It can also be noted that the delay added by the SPSSP varies more than the delay added by the NC repeater functionality.

**D. Results of LAN vs. Public Internet**

This section presents the results for the experimental configuration in Fig. 18. In this setup the NC tool receives synchrophasor information from a PMU connected through a switch in a LAN configuration and from a PMU connected through the public Internet. Both PMUs are of the same type and are equally configured. The PMUs send packets at a 30 fps reporting rate using the UDP protocol. Fig. 19a shows the delays of each PMU as measured by the NC for a duration of 9 hours and 20 minutes. These results show that the average signal delay is much higher for the PMU that sends data through the public Internet. Fig. 19b shows the same delays but for a duration of 10 minutes (within the 9h 20m). The results through the public Internet.

![Fig. 18. Configuration to measure delays from two PMUs: one sending data through the public Internet and the other connected via a LAN (switch).](image)
in this figure show that the delays from the PMU connected through the Internet are also prone to more variability. Fig. 19c shows the probability density functions of the delays from the PMUs in the configuration of Fig. 18 and Table VIII presents statistical information of it. The results in Fig. 19c show that distribution of the PMU coming from the Internet not only has a higher mean value and variance, but is also more right-tailed and is skewed towards the right than the PMU connected through the switch. The results in Table VIII confirm those in Fig. 19c showing that the mean, median and standard deviation are much higher for the case of the PMU data coming from the Internet. The maximum value of delay recorded for the PMU connected via the Internet was 350.97 ms while it was 47.37 ms for the one connected through the LAN. It is important to note that as presented in Table VIII during the duration of this test there were 3 packets lost from the PMU that sends data through the Internet and no packet was lost from the PMU connected via the local network.

This feature allows the tool to analyze the latencies caused by the communication network used for PMU data transfer. A comparison of the tool introduced in this work and the SPSSP also show that the latencies of PMUs are affected by the SPSSP. The results demonstrate that this tool is able to decrease that effect when used instead of the SPSSP.

The paper also presents an analysis of the latencies of a PMU unit for the UDP and TCP protocols using reporting rates of 30 and 60 fps. This analysis shows that the tool presented in this paper performs better than an alternative network analyzer. The results also present a comparison of the delays using a dedicated local area network vs. using the public Internet. This demonstrates the capability of the tool presented in the paper when measuring delays in unreliable networks including congestions and packet losses.

Future applications of this tool include the analysis of the PMU latencies in different network infrastructures, such as in dedicated communication networks, a network with additional TCP traffic and a network with additional UDP traffic. The effect of a PDC used to collect PMU data, secure gateways to encrypt synchrophasor data, and substation PDCs to provide data backup can also be investigated.

TABLE VIII
STATISTICAL RESULTS FOR NC: LAN VS. PUBLIC INTERNET

<table>
<thead>
<tr>
<th></th>
<th>mean (ms)</th>
<th>median (ms)</th>
<th>std dev (ms)</th>
<th>Data dropouts</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAN</td>
<td>43.863</td>
<td>43.87</td>
<td>1.24</td>
<td>0</td>
</tr>
<tr>
<td>Public Internet</td>
<td>107.12</td>
<td>106.92</td>
<td>2.39</td>
<td>3</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS AND FUTURE WORK

This paper introduces a tool, named network characterizer (NC), to accurately measure latencies of synchrophasor data in communication networks. The paper describes in detail the design considerations of such a tool which is composed of two parts: a real-time unit that is connected to a GPS clock for time accuracy and a conventional server for visualization and storing purposes. In addition, the tool introduced in this paper has a repeater functionality similar to the SPSSP.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of: the BPA Technology Innovation Program, project number TIP 289, Dr. Dmitry Kosterev, technical POC, and Mr. Gordon Matthews, PM, with matching funds provided by the U.S. Department of Energy Office of Electricity (DOE/OE) Transmission Reliability Program managed by Mr. Phil Overholt, and the DOE/OE Energy Storage Program managed by Dr. Imre Gyuk.

REFERENCES


Christoph Lackner received the B.S. degree in electrical and computer systems engineering and his M.S. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA in 2015 and 2016 respectively. He is currently pursuing his Ph.D. in Electrical Engineering at Rensselaer Polytechnic Institute. His research interests include power system generation operation and control, and application of synchrophasor data in power systems.

Felipe Wilches-Bernal (S’2012, M’2016) received the M.S. degree in control systems and signal processing from Université Paris-Sud XI, Orsay, France and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA. He joined the Electric Power Systems Research Department at Sandia National Laboratories in 2015 where he currently works as a Senior Member of Technical Staff. His research experience and interests include power system stability and control, renewable energy and smart grid technologies, and signal processing and control techniques applied to power systems.

Brian J. Pierre (S’2009, M’2015) holds a B.S. degree in electrical engineering from Boise State University, Boise, ID, USA; and an M.S. and Ph.D. in electrical engineering from Arizona State University focused on electric power systems, Tempe, AZ, USA. He is presently a Senior Member of Technical Staff in the Power Systems Research Department at Sandia National Laboratories, Albuquerque, NM, USA. His research interests and expertise include power system modeling, power system controls, renewable energy integration, power system resilience, and power system optimization.

David A. Schoenwald (S’1984, M’1992, SM’2002) is a Principal Member of the Technical Staff in the Electric Power Systems Research Department at Sandia National Laboratories. In his current work, Dr. Schoenwald focuses on control system design for damping inter-area power system oscillations for which he received the 2017 R&D 100 Award. In addition, his interests include mitigation of network-induced issues in feedback control systems using wide-area measurements, and development of performance standards for grid-scale energy storage. Before joining Sandia, he was with Oak Ridge National Laboratory, where he developed models and controls for manufacturing applications. He was also an adjunct assistant professor in the Electrical Engineering Department, University of Tennessee, Knoxville, where he taught a class on nonlinear control systems.