Evaluation of Time Gateways for Synchronization of Substation Automation Systems

Paolo Ferrari, Alessandra Flammini, Stefano Rinaldi
Department of Information Engineering
University of Brescia, Via Branze 38, 25123 - Brescia, Italy.
Tel: +39 030 3715445, Fax: +39 030 380014
{paolo.ferrari, stefano.rinaldi}@ing.unibs.it

Gunnar Prytz
ABB Corporate Research
Billingstad, NO-1396, Norway
Gunnar.Prytz@no.abb.com

The IEC 61850 standard is widely used in substation automation system, even if some aspects related to the network-based time synchronization are still under investigation. The latest version of the IEC 61850 standard introduces IEEE 1588 PTP for distributing time in the Station and Process Bus, in addition to the previously used SNTP. Some time synchronization problems may arise when mixing old and new IEC 61850 devices in the same system; the IEEE 1588 PTP and SNTP technologies have somewhat different time representations and synchronization schemes, requiring time gateways for smooth integration. This paper introduces and compares the performance of some compact Time Gateways with different implementation architectures. All the examined gateways have the same external structure: they are transparent, two-port devices which are inserted in the last network link between the switch and the end device, in order to perform the time conversion from IEEE 1588 PTP to SNTP. The Time Gateways are built using the same hardware platform based on an FPGA that enables the creation of real embedded prototypes. The experimental results show that all the considered Time Gateways are applicable to substation automation system, but some of them have better performance than others in term of synchronization accuracy. Moreover, the authors identify the bottleneck in the SNTP implementation of the Time Gateway architecture. A carefully analysis of the behaviour of SNTP is proposed, and useful suggestions for trading off between synchronization accuracy and Time Gateway complexity are given.

Index Terms— IEC 61850, IEEE 1588, SNTP, Boundary Clock, embedded system.
I. INTRODUCTION

Modern Substation Automation Systems (SAS) are generally built as distributed systems based on communication networks. The network infrastructure becomes the reliable backbone that connects all the components of the system. Typically, the systems are based on Ethernet and on the Quality of Service this technology platform provides [1][2]. Modern SAS are designed according to the IEC 61850 standard [3], first released in the 2003 and now being updated, which defines the communication protocols and infrastructures adopted in the practical implementations of the substation. The IEC 61850 also deals with the time synchronization inside SAS, which is the topic of this work. From the functional point of view, the SAS have various requirements on time synchronization, all of them aimed to guarantee that the quality of the power output meets the desired characteristics. The less demanding applications (e.g. fault recorders) require synchronization accuracy in the order of milliseconds, which can be achieved by means of the SNTP (Simple Network Time Protocol) [4] when the systems are relatively small. On the other hand, the complexity and size of modern SAS are growing and high demanding applications (e.g. fault locators) may requires synchronization accuracy in the order of hundreds of nanoseconds.

The described requirements cannot be handled with SNTP which is highly sensitive to network load. SNTP is used to synchronize devices across a network by exchanging messages in a client-server fashion. SNTP is a fairly slow synchronization scheme which is not able to compensate for delays due to network traffic leading to queuing in the end nodes and the network switches (see also Section II.B). An SNTP client requests time information from an SNTP server and calculates the time offset based on the response. However, the client cannot separate between actual transmission delays and the delays caused by queuing from other traffic. Thus, the accuracy is highly dependent on the network size (cascade of switches) and load; in practice accuracies in the milliseconds range can be expected.

The IEEE 1588 protocol [5] (PTP) is a much more robust time synchronization technology with mechanisms to compensate for delays caused by other network traffic. This requires the use of PTP aware switches, i.e. switches with proper support for PTP as a higher layer entity, and that PTP is supported by the end nodes requiring synchronization. When using PTP aware switches, the PTP synchronization accuracy is independent form the network load (at least up to a very high bandwidth utilization) and can be in the microseconds range or better depending on the implementation (e.g. by using hardware assisted timestamping of the receipt and transmission of PTP messages) [6]. Thus, PTP is much better suited for modern SAS than SNTP. For this reason, a specific PTP profile (IEEE C37.238 [7]) for usage in power systems has been also defined.
Some very new instruments and devices for SAS compliant with IEC 61850 embed the PTP functionality. However, there is also the large installed base of IEC 61850 devices without PTP support to take into account. Even if some support for PTP through a software update may be possible, the desired high-accuracy time synchronization could not be achieved because the hardware timestamp unit is generally missing in old devices. If improved synchronization is really needed, or if large networks have to be created, new hardware interfaces must be installed; since most of the industrial products do not support such a hardware update, a full replacement of the system should be considered. In practice, full replacement is not realistic on the short term due to the high cost of SAS components. As a result, the already installed SAS may continue operating with devices without PTP support (i.e. with reduced synchronization accuracy) for quite a long time.

Technically, PTP and SNTP devices can operate independently even if connected to the same network infrastructure. However, in order to obtain the best synchronization performance from the PTP devices, the network infrastructure has to be “PTP aware”; in other words, all the network devices, i.e. the switches, must provide accurate time synchronization independent of network load, propagating the correct notion of time to the end-nodes by means of “boundary clock” or “transparent clock” approaches described by IEEE 1588. If a network infrastructure is “PTP aware”, the PTP frames are handled in a special way in order to reduce the influence of network traffic load on their propagation delay (see Section II.B). Other types of traffic, including the SNTP packets, are influenced by the network traffic. Therefore, the synchronization accuracy of the SNTP protocol is not enhanced by a network with “PTP aware” infrastructure. As shown in [8], the coexistence of old and new devices that use these synchronization protocols on the same network may be improved using dedicated devices, the Time Gateways, which are able to translate the time from the PTP domain to the SNTP domain in the last network link before the end device. By the way, the use of the Time Gateways approach also guarantees an easy scalability of the network without appreciable reduction of the synchronization accuracy.

In this paper, which is the continuation of [9], the time synchronization requirements of a Substation Automation System are discussed referring to both the old and new synchronization solutions included in the IEC 61850 standard. The general structure of the Time Gateway is introduced, and the strategies for propagating the time information from the new PTP-compliant infrastructure to the old SNTP clients (the end-users of the time information) are analyzed. The general aim of the paper is to experimentally evaluate the synchronization performance of the different architectures used for the Time Gateways, discussing mainly the tradeoffs between a full-software (low cost and complexity) and a hardware (high-accuracy) implementation.
The paper is structured as follows: In Section II, an overview of the technologies involved in SAS is presented, focusing the attention on time distribution requirements and solutions. The Time Gateway architecture, developed to provide time information to SNTP devices in an IEEE 1588 network [8]-[10], is briefly introduced in Section III. Three Time Gateways are implemented using different strategies, each of them with a specific behavior. The experimental evaluation of the synchronization performance of the three systems is provided in Section IV; the different implementations are compared and the SNTP behavior is analyzed and discussed in detail. Finally, the results are summarized in the conclusion.

II. THE TECHNOLOGIES INVOLVED IN SAS

A. IEC 61850

The IEC 61850 standard for communication systems in power substations [3] provides abstract definitions of services and data independent from the underlying protocols by means of an object model description for equipment and functions. IEC 61850 organizes the electric SAS into three logical levels: the Process Level consisting of the switchyard apparatus, such as remote I/Os, actuators and current and voltage transformers for measurements; the Bay Level hosting the protection and control IEDs (Intelligent Electronic Devices) and the Station Level including the functions operating on more than one Bay and the coordination with the control center. The different levels are connected by means of two buses: the Process Bus connects the Process and the Bay Levels, carrying time critical communication; the Station Bus links the Bay Level with the Station Level, carrying both management and time critical information. The IEC 61850 standard maps the services and the data structures on to Ethernet.

Several SAS applications require additional synchronization services to work properly. The data collected by a distributed measurements system are affected by the communication and synchronization uncertainty [11]. Fault recorders and fault locators [12] are examples of applications benefitting from accurate time synchronization. The IEC 61850 provides for different time synchronization classes, each of them specifies the required accuracy. In Table I, the time synchronization classes defined in IEC 61850-5 ed. 1 are compared to the new classes defined in IEC 61850-5 ed. 2. Note that the latest release of IEC 61850 introduces the additional TL and T0 classes, dedicated to applications requiring synchronization accuracies in the range from 10 ms and upwards (towards lower accuracies). A one milliseconds synchronization accuracy is usually enough for data timestamped by the SCADA system on the station bus. On the other side, the merging units on the process bus need to be strictly synchronized, usually with a maximum sampling error in the order of 4 µs, as defined in IEC 61850-5. To satisfy this requirement, the 61850-9-2LE [13] suggests the use of a dedicated Pulse Per Second signal (1-PPS). The measure of the
ynchrophasors, defined in the IEEE C37.118 standard [14], does not specify a maximum time synchronization error, though to satisfy the maximum measurement error, called Total Vector Error (TVE), a synchronization accuracy in the order of a few microseconds is usually needed [15].

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>IEC 61850 TS ed.1</th>
<th>IEC 61850 TS ed. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;±10 ms</td>
<td></td>
<td>Class TL</td>
</tr>
<tr>
<td>±10 ms</td>
<td></td>
<td>Class T0</td>
</tr>
<tr>
<td>±1 ms</td>
<td>Class T1</td>
<td>Class T1</td>
</tr>
<tr>
<td>±0.1 ms</td>
<td>Class T2</td>
<td>Class T2</td>
</tr>
<tr>
<td>±25 µs</td>
<td>Class T3</td>
<td>Class T3</td>
</tr>
<tr>
<td>±4 µs</td>
<td>Class T4</td>
<td>Class T4</td>
</tr>
<tr>
<td>±1 µs</td>
<td>Class T5</td>
<td>Class T5</td>
</tr>
</tbody>
</table>

Table I: Comparison of IEC 61850 ed.1 and ed.2 time synchronization classes.

B. Synchronization protocols in communication network

The use of a unique time reference across all the SAS is very important to manage complex tasks and to monitor the behavior of the substation (e.g. alarms and protections). The de-facto standard for time synchronization in a network is the Network Time Protocol (NTP) [16]. This protocol is used to synchronize clocks over a packet switched network with variable propagation delay, such as Internet. The algorithms adopted allow to achieve a time synchronization accuracy in the order of milliseconds over the Internet and in the order of hundreds of microseconds on a local network: in practice, the time synchronization accuracy depends on the traffic load of the network, the size of the network and the characteristics of the nodes that are being synchronized (e.g. the real-time properties of the nodes and how the NTP protocol is implemented) [17]. A less complex implementation of NTP, that uses the same protocol but without some of the complex time-tracking algorithms, is known as the Simple Network Time Protocol (SNTP). Usually, it is used in embedded devices since it requires fewer resources than NTP and in applications where high accuracy timing is not required. The IEC 61850-8-1 provides for an SNTP based synchronization system. However, this protocol is not able to satisfy the strictest synchronization classes defined in IEC 61850-5, as well known in literature [18]. The synchronization performance of the SNTP protocol can be improved using specific techniques, such as the timestamping of SNTP packet at the hardware level [19],[20]; however these solutions are not compatible with the IEC 61850 SNTP clients available [21], limiting the feasibility of this approach. Usually, in current SAS implementations, a dedicated point-to-point synchronization network is adopted to distribute the time information, using the
IRIG-B time code, from the GPS receiver to the end-devices. This solution can satisfy also the most strict time requirements but it requires the use of a parallel dedicated network, which has to be often calibrated to compensate environmental effects.

The IEEE 1588 [5] protocol, also known as the Precision Time Protocol (PTP), is a protocol used for time distribution over local area networks, especially those based on Ethernet, like industrial real-time networks [22]. Compared with NTP, it provides accuracy below the microseconds when dedicated hardware supports the protocol stack [23][24]. The protocol organizes the clocks of the local network into a master-slave structure: the “best” clock of the network is elected as master by means of the Best Master Clock (BMC) algorithm. The PTP standard has different profiles, each of them dedicated to a specific application; for instance, the IEC 61850 standard has recently adopted a PTP profile for power systems [7] to synchronize the IEDs in electrical automation systems. It should be noted that various PTP profiles may have significant differences which may create compatibility issues for equipment with different profiles implemented.

It should be highlighted that SNTP and PTP use different approaches to distribute the time information. NTP/SNTP has a Server/Client architecture, in which a Client recovers the time from multiple servers available with non-periodic requests. On the other hand, the PTP protocol provides for a Master/Slaves approach, in which the Master periodically sends synchronization messages to the slaves via unicast or multicast messages depending on the specific profile. The power system profile requires the use of multicast messages. The time scales adopted by the two protocols are different: the NTP epoch was January 1st, 1900, whereas the PTP epoch was January 1st, 1970. Moreover, PTP, with respect to NTP/SNTP, allows mapping the synchronization messages on different communication layers (mainly layer 2 Ethernet, and layer 3 UDP) as specified in the different profiles available. A switched Ethernet infrastructure introduces a variable delay on the propagation time of synchronization packets, affecting the synchronization performance. The IEEE 802.1Q priority, when supported by the overall infrastructure, can be used to reduce the delay of high priority packets, thanks to the management of queues in the switch. However, also in this case, a 10/100 Mbps store and forward switch may introduce a variable delay in the order of 120 μs (i.e. the maximum duration of an Ethernet packet). A PTP-aware network infrastructure, obtained using PTP Boundary Clocks as well as PTP Transparent Clocks, handles PTP messages in a special way: the delay introduced by switch queuing is compensated using hardware timestamping and frame modification, allowing accurate time propagation also in a busy network infrastructure. Usually, in real-time Ethernet networks, where IEEE 1588 is widely accepted, the IEEE 802.1Q priorities are strictly managed and a high priority is assigned to the most critical messages of the control application. Note as, in the case PTP-aware switches are adopted, the priority of the PTP messages is not so important since the propagation delay is compensated by the device itself. Under these conditions, the SNTP Client/Server synchronization is not only affected by the
traffic load of the network, but also by the presence of higher priority messages, decreasing the performance of the protocol. In this scenario, the PTP aware switches are able to transfer accurate time information to local PTP enabled devices. By means of a gateway device, able to recover the PTP time from the PTP infrastructure and to translate that time into the SNTP time scale, it is possible for the two time synchronization protocols to work together, improving the synchronization accuracy for SNTP-only enabled devices.

III. COMPARISON OF POSSIBLE ARCHITECTURES FOR PTP TO SNTP TIME GATEWAY

A typical application scenario is shown in Figure 1; the Time Gateway provides the SNTP time to an old device placed into an IEEE 1588-aware network [9]. From the IEEE 1588 point of view, the proposed device is a special Boundary clock formed by two ports: it works as a PTP slave on the side connected to the PTP domain, recovering the time from the infrastructure and correcting the local clock, whereas it works as an SNTP server on the other side, providing the local time to the SNTP client.

This type of Time Gateway could be deployed in modern substation systems that have to be updated or extended, as discussed in [9]. In such cases, even if the time synchronization requirements have not increased, it may be difficult to fulfill the requirements for old devices with the use of SNTP when the network gets bigger and more complex, since every hop introduces additional variability to the propagation of the synchronization messages, as experimentally demonstrated in Section IV.A. The Time Gateway may thus be used to increase or at least maintain synchronization accuracy in old devices, avoiding the replacement of the whole system.

Figure 1: A typical application scenario of the Time Gateway applied to a Substation Automation System.

The Time Gateway, as discussed in [8], should ideally be an easy plug and play device transparent to all protocols used in the network (including configuration and management tools but excluding PTP and SNTP), which can be connected directly on the cable.
The IEC 61850 standard suggests considering 1 ms as the overall synchronization accuracy on the “Station bus” of a Substation. In order to fulfill such a constraint in large Substation Automation Systems, the target synchronization accuracy for the Time Gateway may be roughly set below 500 µs. However this is not a strict requirement, and in some conditions, it is enough just to maintain the current performance.

The architecture of the Time Gateway is reported in Figure 2. The embedded system uses the µClinux OS [26], kernel 2.6.30, a Linux kernel dedicated to embedded processors. The PTP synchronization stack has been implemented using the PTPd [27] daemon (the source code of PTPd is freely available [28]). PTPd supports both PTP protocol versions (v2002 and v2008) and it can support, with some programming effort, the different IEEE 1588 profiles. It should be noted that the different PTPd versions and profiles are not necessarily interoperable, so the Time Gateway has to be able to adapt to the PTP version/profile of the network infrastructure. Since the Time Gateway has been developed for SAS networks, the predefined profile is [7]. The PTPd daemon receives the sync messages sent by the PTP master through the PTP port (see Figure 2). Originally, the PTPd is a software-only PTP implementation, i.e. the PTP messages are timestamped at kernel level; these timestamps are then used to estimate the time offset from the Master clock and to adjust the local clock. This approach can be enhanced in several ways; each of them requires increased effort and additional cost. In the following, two such improvements are introduced and compared.

![Figure 2: The proposed approach for the Time-Gateway based on an in-line boundary clock.](image)

The first improvement requires a brief discussion about the PTPd control loop algorithm. The IEEE 1588 standard does not specify a control loop algorithm, since it could depend on the performance and application requirements. For example, the PTPd adopts a PI loop and simple moving average filters, enough to obtain a synchronization in the order of tens of microseconds. The experimental analysis of the timestamp mechanism [10] highlights that a simple moving average filter is not enough for an accurate synchronization because of impulsive noise introduced by the OS scheduling. In order to reject the outliers, the filtering stage of the PTPd has been improved by adding a median filter on the estimated offset (10 samples), as discussed in [8].
The second improvement has a greater impact on performance. The synchronization accuracy of the PTPd can be further improved by means of the dedicated Ethernet PHY used in this design (DP83640 from National Semiconductor), which is able to assist the PTPd with its dedicated hardware features (Figure 3). The Linux kernel can support this new class of devices thanks to recently included patches [29]-[30]. As further improvement, the System Time of μClinux and the PTP time of the DP83640 may be syntonized since the FPGA is clocked with the clock signal generated by the PHY. The offset between PTP and the System Time is continuously monitored and should be constant.

On the SNTP side, the same approach has been used in all the implementations. The MSNTP Server (an open source SNTP server) responds to SNTP client requests forwarding the local System Time, which is corrected using the measured offset.

Regarding the other network traffic, which is not related to time synchronization, μClinux has been compiled to support the 802.1D Ethernet Bridging and the bridge-utils package has been added. In this way, it is possible to create a Linux bridge device with two physical ports (PTP port and SNTP port) and the Time Gateway thus also works as a software bridge (layer 2 device), whose performance is adequate for a proof-of-concept demonstrator.

![Figure 3: The detailed block diagram of the Time-Gateway.](image)

IV. EXPERIMENTAL RESULTS

The Time Gateway prototype has been realized using a NIOS II development kit from Altera equipped with the EP2S60F672C3 Stratix II FPGA (60k LE). The FPGA development board provides a 10/100 Mbps Ethernet MAC (SMSC LAN91C111), 32 MB DDR SDRAM, and 16MB Flash. An external evaluation board (DP83640T-EVK), equipped with a PTP-aware 10/100 Mbps Ethernet PHY (DP83640), has been connected to the main FPGA board. The prototype requires 13% of the logic and less than 5% of the memory available on the FPGA. The device firmware including the OS, the required drivers and applications, occupies less than 2 MB. More information about the implementation is provided in [8]-[9].
In the following, a reference system with SNTP is evaluated as a term of comparison. Then the different implementations of Time Gateway described in the previous section are experimentally evaluated and compared, highlighting the tradeoff between the synchronization performance and design effort.

A. Evaluation of SNTP messages propagation over a IEC 61850 station bus

In this section an experimental analysis of the propagation delay of SNTP messages over the typical network topology of a station bus is presented. This experimental analysis allows highlighting the decrease of the SNTP performance due to the network infrastructure size and load.

The IEC 61850-90-4 suggest some network topologies for increasing the availability of the station bus. One of the commonly used is the ring topology since it is easy to create adding a single link. However, ring architectures typically increase the number of hops between the SNTP server, usually connected to a GPS receiver close to control station, and the SNTP clients, installed in the bay. Since the SNTP off-the-shelf components do not support hardware timestamping or the compensation of the delay introduced by the switches, each hop between the SNTP server and the clients affects the synchronization accuracy.

In an SNTP system, at the synchronization moment \( k \), the Client requests (at time \( T_{C1}(k) \)) the time from a Server using a dedicated message (SNTPrequest). The Server responds with a response message (SNTPresponse) that contains the Server reception time of the SNTPrequest (\( T_{S2}(k) \)) and the Server transmission time of the SNTPresponse (\( T_{S3}(k) \)). The response from the Server is received by the Client at time \( T_{S4}(k) \). The messages exchanged between the SNTP client and server are shown in Figure 4.b. These timestamps are used to estimate the roundtrip delay (\( \delta(k) \)), i.e. the time it takes for a message to reach the Server and then to go back to the Client, and the offset (\( \theta(k) \)), i.e. the difference between the Server reference time and the Client local time:

\[
\delta(k) = \frac{T_{S4}(k) - T_{C1}(k) - T_{S2}(k) + T_{S3}(k)}{2} + \frac{T_{S3}(k) - T_{C4}(k)}{2} \quad (1)
\]

\[
\theta(k) = \frac{T_{S4}(k) - T_{C1}(k) + T_{C3}(k) - T_{S2}(k)}{2} - \frac{T_{S3}(k) - T_{C4}(k)}{2} \quad (2)
\]

Generally speaking a typical station bus infrastructure is formed by a ring of Ethernet switches, managed by the RSTP protocol, as the one described in [25]. The network architecture considered in this section, after the configuration of RSTP, is shown in Figure 4. The SNTP server (the GPS receiver Meinberg LANTIME M600), is connected to a side of the ring, while the SNTP client is connected on the other side of the station bus, at the end of the line, after four 10/100 Ethernet switches.
Two Ethernet taps (NetOptics 10/100 Tap) are connected on the two sides of the network to monitor the traffic without affecting the propagation of the messages. A network analyzer (Endace Ninjabox) is used to log the traffic on the station bus on the two sides of the network. This device is able to assign an accurate timestamp to each packet on the network (timestamping uncertainty below 10 ns). In ideal conditions (i.e. no traffic on the network), the propagation delay of the SNTP messages can be easily estimated and depends only on the number of hops between the client and the server, and the bridge delay introduced by each switch, which depends on the technology adopted. In these conditions, the SNTP synchronization is limited only by the uncertainty introduced by the timestamping method used by the SNTP application. In a more realistic scenario, the traffic load on the network affects the propagation of the SNTP messages. Thus, a packet generator is used to inject traffic on the station bus (10% of available bandwidth, i.e. 10 Mbps) in the direction from the SNTP server to the client, to analyze the system under realistic condition. 80% of the traffic is formed by packets of 200 bytes, the rest of packet of 500 bytes. The SNTP client sends every 2 s a SNTPrequest to the SNTP server. The network analyzer logs and timestamps the SNTP traffic on the two sides of the network. The time difference between the timestamps assigned to a packet at the two sides of the network is a measure of its propagation delay. The distributions of the delay of the SNTP messages (4000 samples) in the two directions are reported in Figure 5. Note that the traffic on the network adds a variable propagation delay to the SNTP messages. The SNTPresponse messages (from the server to the client, shown in Figure 5.a) may experience a larger delay compared to the SNTPrequest in the other direction (Figure 5.b), due to the generated traffic. Since the distribution of the propagation delay is not Gaussian, the mean and the standard deviation are not meaningful in practice. In this case, a better estimator is the jitter, which is defined as:

$$jitter = \max_{k=0}^{N-1} \left| \text{delay}(k) \right|$$  \hspace{1cm} (3)

where N is the number of samples obtained during the test, and delay(k) is the propagation delay estimated at the synchronization time k. The jitter of the propagation delay in this experimental setup is in the order of 170 µs.
The distribution of the propagation delay of SNTP packets over a Station Bus, a) from the server to the client and b) vice versa.

As clearly highlighted by (2), the estimation of the offset depends on the round trip delay the SNTP messages experience over the network. An increase in the variability of the $\delta$, due to the traffic on the network or to a change in the topology of the network, causes a decrease in the estimation of $\theta(k)$, limiting the synchronization performance of the SNTP system. The round trip delay estimated during the experiment is shown in Figure 6 (only the first 200 samples are reported for clarity in Figure 6.a). Note that during the experiment the estimation of the round trip is severely affected by the traffic as demonstrated by the highly variable roundtrip delay in Figure 6.b.

It should be remarked that, on the other hand, in a PTP system, the PTP aware switches are able to compensate for the propagation delay of the PTP sync messages, eliminating this source of inaccuracy.

The synchronization performance of the system has been evaluated using the experimental setup shown in Figure 7. A Class B LXI device (TriggerBox E5818A from Agilent) is used as a PTPv1 master, sending a synchronization message every 2 s. The Time Gateway device is able to recover the time from the master through the PTP protocol. The LXI instrument is directly
linked to the Time Gateway in order to reduce the influence of the network traffic on the synchronization estimation. For the same reason, no additional network traffic is present on the network.

The PTPd daemon on the Time Gateway, records the measured time offset and drift in a log file (PTPLog in Figure 7). Three different implementations of PTPd have been compared. The first is the original PTPd as used in Linux systems, i.e. it is the full software implementation of PTP that timestamps the messages at the kernel level. The second is the optimized version of the PTPd developed by the authors for application in Substation Automation system [8], as discussed earlier it is still a full software implementation but introduces the median filter to eliminated sporadic errors. The third is the PTPd modified for using the hardware support. The results of the three versions are compared in Figure 8.

The time offset, i.e. the time difference between the Master and Time Gateway local time, logged during the experiment, which lasted for more than 2 hours, i.e. 4000 samples, is reported in Figure 8.a. Note that only the first 1800 samples of the experiment are reported in the graphs for clarity reasons. In this case, the synchronization capabilities of the Time Gateway are severely affected by the timestamping uncertainty introduced by the OS.

As mentioned in Section III, the filtering stage of the PTPd has been modified inserting the median filter (10 samples) applied to the received timestamps. In Figure 8.b, the synchronization results obtained using the improved PTPd is reported. Note that the adoption of the median filtering stage improves the synchronization accuracy approximately one order of magnitude. The distribution of the time offset (4000 samples) is shown in Figure 8.b. The mean value is 3 µs and the standard deviation is 4 µs. However, since the distribution of the time offset is not Gaussian due to the noise added by the OS discussed above, a better estimator of the synchronization accuracy is the synchronization jitter, defined as:

\[
\text{jitter} = \text{MAX}_{k=0}^{N-1} |\theta(k)|
\]  

(4)

This parameter represents the maximum time offset of the Time Gateway from the Master observed during the experiment.

![Figure 7: Experimental setup to evaluate the performance of the hardware assisted Time Gateway.](image)
Figure 8: Time offset trends and distributions using: a) original PTPd, software only; b) optimized PTPd, software only; and c) hardware-assisted implementation.

The synchronization jitter obtained using the improved PTPd (Figure 8.b) is 11 µs, in accordance with other results reported in literature for a software only PTP implementation [27].

The use of the dedicated Ethernet PHY, which supports the PTPd with hardware timestamp (hardware-assisted PTPd), improves the synchronization accuracy, as clearly shown in Figure 8.c. In this case, the synchronization jitter is 46 ns, two orders of magnitude less than in the results obtained in the previous test case. The distribution of the time offset is shown in Figure 8.c. The mean value is 1 ns and the standard deviation is 18 ns.
C. Time Gateway performance assessment

In the second experiment, the performance of the Time Gateway in distributing the time information using the SNTP protocol has been analyzed. As explained in Section III, the Time Gateway is able to respond to SNTP requests coming from an SNTP Client. In this experiment, the SNTP Client has been implemented using a PC with a Pentium Dual Core 3 GHz CPU, 2 GB RAM, Linux kernel 2.6.37 and SNTP version 4. The SNTP client sends a time request to the Time Gateway every 2 s. The Client adjusts its local clock (calling the adjtime() system function) and logs its time offset from the Time Gateway, the SNTP Server. During the experiment, the System Time of the Time Gateway is synchronized to the PTP Master, the LXI device described in the previous experiment, using the PTPd daemon. The test is repeated using the normal as well as the optimized (software only) PTPd implementation. The Time Offset measured in the two cases by the SNTP Client (for more than two hours, about 4000 samples) is reported in Figure 9.a and in Figure 9.b, respectively, together with the relative distributions (over 4000 samples). For readability purposes, only the first 1800 samples of the experiment are reported. Note that the optimized PTP synchronization increases the overall performance of the system: in the first case, the synchronization jitter is approximately 1450 μs, whereas in the second case it is approximately 400 μs. The results depends on both client and server implementation and may be further improved. However, in the second case, they are acceptable for application in Substation Automation System.
D. Assessing SNTP Server performance

As clearly shown by the experiment in Section IV.C, additional improvements in the PTP synchronization capability of the Time Gateway does not improve the overall performance of the system. Therefore, a detailed analysis of the SNTP capability of the proposed system is required to highlight the limits of this approach.

Generally speaking, the time offset $\theta(k)$ (2) measured by an SNTP Client is affected by the variability introduced by the propagation of SNTP messages over the network and by the timestamping technique adopted by the Server, as well as by the Client. In order to eliminate the first contribution, already evaluated in section IV.A, the SNTP Client is directly connected to the Time Gateway via a cable: the propagation delay of SNTP messages on this link can be considered constant. Therefore, the estimation of time offset is mainly affected by the measurement noise due to the timestamping mechanism adopted. Let $\sigma_C$ be the uncertainty associated with the timestamp provided by the Client and $\sigma_S$ the uncertainty of the Server timestamp. Considering the algorithm used to estimate the offset (2), the uncertainty associated with the estimated offset ($\sigma_\theta$), as specified in [31], is:

$$\sigma_\theta^2 = \frac{1}{2} \sigma_C^2 + \sigma_S^2$$

(5)

The timestamp mechanism adopted in the Server and in the Client is rather similar (software based): it is expected that they provide a similar contribution to the uncertainty of the estimation of the offset. Therefore it is rather difficult to characterize the capabilities of the Time Gateway to distribute the time information independently from the specific implementation of the Client. For this reason, the experimental setup shown in Figure 10 is used to perform a more accurate analysis of the performance of the SNTP Server of the Time Gateway. During this test, the SNTP Client is the same PC of Section IV.C that periodically requests the time (every 2 s) from the Server. The SNTP messages are acquired by a Network Analyzer (NA) through an Ethernet Tap, in order to avoid any perturbation of the message propagation time on the network. The Network Analyzer is able to timestamp the SNTPrequest and SNTPresponse packets using a dedicated hardware timestamping unit (HW TSU). Using these timestamps ($T_{NA1}(k)$ and $T_{NA4}(k)$), and the timestamps provided by the Server, is possible to estimate the time offset of the network analyzer ($\theta_{NA}(k)$) as:

$$\theta_{NA}(k) = \frac{T_{NA1}(k) - T_{NA0}(k)}{2} + \frac{T_{NA4}(k) - T_{NA4}(k)}{2}$$

(6)

However, the $\theta_{NA}(k)$ is affected by the drift between the clock of the network analyzer and the SNTP server. In order to compensate the terms due to the drift of the clock, the time offset can be re-written as:
The uncertainty ($\sigma_{\delta_{\text{SNTP}}}$) in the estimation of the offset $\hat{\delta}_{\text{SNTP}}(k)$ is mainly due to the measurement uncertainty of the timestamping mechanism of the Server ($\sigma_S$), since the measurement uncertainty of the timestamping mechanism of the Network Analyzer ($\sigma_{\text{NA}}$) is negligible (< 10 ns [23]). The compensated time offset estimated using the network analyzer ($\hat{\delta}_{\text{SNTP}}(k)$), measured over more than two hours (4000 samples), is reported in Figure 11. As discussed above, the measurement uncertainty of the time offset reported in the figure is an index of the capabilities of the SNTP Server to distribute the time, independently from the performance of the Client, i.e. it is a measure of the best performance that can be obtained. The maximum time offset measured during the experiments is of the order of 300 $\mu$s. Hence, it is expected that an ideal Client cannot be synchronized with a jitter (4) better than 300 $\mu$s.

![Figure 10: Experimental setup for assessing SNTP Server performance.](image)

![Figure 11: Compensated time offset a) obtained using the hardware timestamp provided by the NA and b) distribution (4000 samples).](image)

The limit of the current approach is due to a mix of different factors. The SNTP protocol does not have the Follow_up message (as in the PTP case) for transferring the precise transmit timestamp of the SNTP response message (obtained in the kernel space) to the Client. The only available timestamp is obtained in the OS user space. In the current implementation,
µClinux is not a real-time OS, therefore any operations in user space (such as the handling of timestamps) can be randomly delayed by several hundreds of microseconds.

V. Conclusions

The mixed use of old and new devices in IEC 61850-based SAS may be impaired by problems with time synchronization; old devices use SNTP and NTP, while more recent devices may use PTP. In order to solve these problems, the authors proposed in previous papers the use of a Time Gateway which is able to transparently translate time from the PTP domain to the SNTP domain. The translation is done only in the last link between the PTP compliant infrastructure and the old SNTP device, improving performance with respect to a network where PTP and SNTP work in parallel.

In this paper, three implementations of the Time Gateway are proposed and experimentally evaluated by means of real prototypes. The Experimental results show that the considered architectures are able to guarantee an overall synchronization jitter from 10 µs down to 50 ns at the PTP side depending on the implementation. The maximum jitter at the SNTP side in the last link is greater, about 400 µs. Further experimental investigations identified the SNTP server as responsible for the performance reduction. As a conclusion, the overall performance is limited by the implementation of the SNTP side of the Time Gateway; since SNTP does not use hardware timestamp, the synchronization accuracy limit is set by the software execution jitter. As a matter of fact, having a hardware assisted PTP implementation does not increase overall performance, unless the SNTP side is hardware assisted too.

However, the results obtained with all the considered Time Gateways are more than acceptable for many typical IEC 61850 applications. The Time Gateway may constitute a valid alternative to the full replacement of an old device, when high synchronization accuracy is required or large networks must be implemented.

REFERENCES


