Understanding and Applying Precision Time Protocol

Steve T. Watt, Shankar Achanta, Hamza Abubakari, and Eric Sagen
Schweitzer Engineering Laboratories, Inc.

Presented at
Saudi Arabia Smart Grid 2015
Jeddah, Saudi Arabia
December 7–9, 2015

Previously presented at the
14th Annual Clemson University Power Systems Conference, March 2015

Originally presented at the
Power and Energy Automation Conference, March 2014

Understanding and Applying Precision Time Protocol

Steve T. Watt, Shankar Achanta, Hamza Abubakari, and Eric Sagen, *Schweitzer Engineering Laboratories, Inc.*

Abstract—**Precise time synchronization has become a critical component of modern power systems. There are several available methods for synchronizing the intelligent electronic devices (IEDs) in a power system. In recent years, there has been great interest in providing time to the IEDs using the same infrastructure through which the data are communicated. Precision Time Protocol (PTP) is a promising technology for achieving submicrosecond synchronization accuracy between IEDs over Ethernet. This paper presents the protocol fundamentals and discusses considerations for designing power system networks to achieve submicrosecond accuracies. Specific provisions made in the profile for power system applications to support IEC 61850 substation automation systems are also discussed.**

I. Introduction

Many intelligent electronic devices (IEDs) in modern power systems execute distributed applications that rely on common and precise time between IEDs. Synchrophasors, Sampled Values (IEC 61850-9-2), and traveling wave fault location are some application examples that require submicrosecond time synchronization accuracy. Ethernet, which started out as a communications technology for computers, is rapidly becoming a dominant communications technology for IEDs in substations and industrial environments. Because Ethernet is a packet-based technology, the challenge with using it in substations is often about achieving high reliability, determinism, and availability [1]. Traditionally, the dominant method of distributing precise time at substations has been IRIG-B, which requires a separate cable to the IEDs in addition to the Ethernet or serial cable used to communicate the application data. The IEEE 1588 Precision Time Protocol (PTP) distributes precise time with better than 1-microsecond accuracy over Ethernet, which is becoming the standard technology for IED communication.

Network Time Protocol (NTP) is the most widely used time synchronization protocol in the world. Almost every computer connected to the Internet is time-synchronized by NTP. All personal computers (PCs) running Windows® or Linux® come with NTP time synchronization. Computers and other communications devices in a substation benefit from the ease of using NTP to set the local time in these devices, where accuracies of subsecond synchronization are acceptable. NTP is distributed through Ethernet-capable devices.

NTP uses a client-server model for the communication of time information between devices. The function of a server is to provide accurate time to its clients. The individual clients run a small program as a background task that periodically queries the server for accurate time information. The frequency of these queries is generally about 15 minutes in order to maintain the synchronization accuracy for the network.

Table I compares IRIG-B, NTP, and PTP time synchronization protocols [2].

TABLE I
COMPARISON OF TIME-DISTRIBUTION METHODS

Time-Distribution Methods	IRIG-B	NTP	PTP (IEEE 1588 and IEEE C37.238)
Physical Layer	Coaxial cable	Ethernet	Ethernet
Model	Master-slave	Client-server	Master-slave
Synchronization Accuracy	~100 ns to 1 µs	~1 to 100 ms	~100 ns to 1 µs
Compensation for Latency	Yes, using cable length as user input	Yes	Yes
Update Interval	Once per second, pulse per second (pps)	Minutes	Configurable, typically once per second
Hardware Requirements	Special hardware required at master and slave	Master only	Hardware support required for high accuracy
Relative Cost	Medium (IRIG-B cabling)	Low (software)	Medium to high (early adoption)

In order to achieve the submicrosecond accuracy and precision specified in the PTP standard, a thorough understanding of the protocol and adherence to specific design principles are needed. It is especially important to understand how network asymmetry, message delay variations, and network topology affect accuracy and precision.

This paper presents the fundamental principles and operation of PTP and discusses considerations for designing power system networks to achieve the promised submicrosecond accuracy.

II. Precision Time Protocol

IEEE 1588 PTP is a message-based time transfer protocol that enables synchronization accuracy and precision in the submicrosecond range for packet-based networked systems [3]. The standard was first released in 2002 as Version 1 and then revised in 2008 as Version 2. The two versions are not compatible; therefore, it is not possible to have a mix of Version 1 and Version 2 devices in the same network. This paper focuses entirely on Version 2 because it is the most widely deployed version.

A. PTP Device Types

PTP defines the following five device types: ordinary clocks, boundary clocks, end-to-end transparent clocks, peer-to-peer transparent clocks, and management nodes.

An ordinary clock is a device that either serves time or synchronizes to time and communicates on the network through a single PTP port (Ethernet interface). An ordinary clock is called a grandmaster clock if it is serving time to the entire PTP network and therefore is the ultimate source of time for all other devices in the network. An ordinary clock is called a slave clock if it synchronizes to another clock serving time (such as a grandmaster or boundary clock). The IEDs that run the application algorithms are typically slave-only clocks.

A boundary clock is a multiport network device that synchronizes to the reference time on one port and serves time on one or more ports. That is, one of the ports is a slave port and the rest of the ports are master ports. In essence, boundary clocks terminate and then start the time distribution. This functionality is usually built into PTP-aware network components, such as switches, bridges, and routers. Boundary clocks can be used to scale up a PTP network by servicing requests from slave clocks that would otherwise be serviced by the grandmaster clock. This makes it possible to support a large number of slave clocks (IEDs) in the network.

An end-to-end transparent clock is a multiport network device that measures the length of time a PTP message spends within the device as it is routed from the ingress port to the egress port and then adds that information to a correction field in the message. This is intended to eliminate any variations in message delays and asymmetry that the device may introduce in the transfer of PTP messages. The end-to-end transparent clock functionality is typically performed by PTP-aware switches.

A peer-to-peer transparent clock is a multiport network device that measures the link delay of each port and adds that information and the residence time to PTP messages traversing the device. Like the end-to-end transparent clock, the peer-to-peer transparent clock eliminates asymmetry and packet delay variations in the device. Additionally, it allows for scaling because slave devices do not have to send requests to the grandmaster clock to measure the end-to-end delay. Instead, each device measures the delay to its peer (i.e., the peer-to-peer delay). Fig. 1 shows the relationship between the end-to-end and peer-to-peer delay measurements. The peer-to-peer delay measurement is especially suited for networks with redundant paths because the PTP message will always contain the actual delay it experiences on the network, regardless of the path it takes. With the end-to-end delay measurement, a computed delay for one path can be used for offset (time error) computations for a message that traversed a different path with a different delay.

A management node is a network-connected device used to configure and monitor PTP devices. It is typically a computer. A nontransparent switch is a device that does not support PTP and does not account for the residence time for the traffic going through it.

Fig. 2 shows an example PTP network topology.

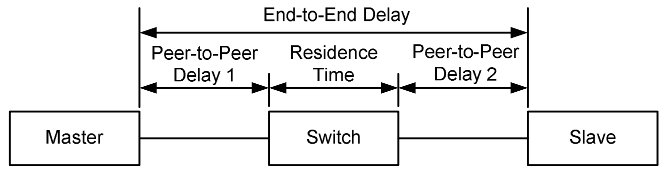

Fig. 1. Relationship between end-to-end and peer-to-peer delay measurements

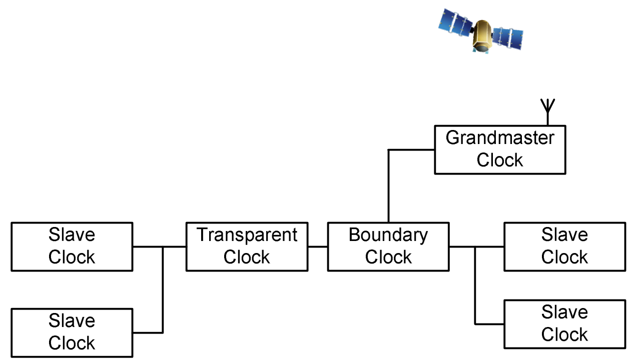

Fig. 2. Example PTP network topology

B. PTP Basic Operation

PTP defines a number of protocol messages and classifies each as an event message or a general message. An event message is a message that must be accurately time-stamped at the time of transmission, reception, or both. The event messages are *Sync*, *Delay_Req*, *Pdelay_Req*, and *Pdelay_Resp*. A general message does not need to be time-stamped. The general messages are *Announce*, *Follow_Up*, *Delay_Resp*, *Pdelay_Resp_Follow_Up*, *Management*, and *Signaling*. Management messages are used by management nodes to configure and retrieve configuration information from PTP devices. Signaling messages are used by PTP clocks to negotiate certain optional services, such as unicast transmission. PTP uses multicast communication by default.

The operation of PTP is conceptually a two-stage process. In the first stage, the PTP clocks are self-organized into a hierarchy in which the grandmaster clock with the highest priority and best quality is at the top and the slave clocks are at the bottom. Boundary and transparent clocks exist in the middle of the hierarchy. In the second stage, protocol messages are exchanged to synchronize all clocks ultimately to the grandmaster clock.

The best master clock algorithm (BMCA) is used to organize the clocks into a hierarchy and to let slave clocks use the best (most precise) time available on the network. Ports on ordinary clocks (except slave-only clocks) and boundary clocks transmit *Announce* messages containing the clock priority and quality. Each clock on the network is able to use the BMCA and the clock properties received from *Announce* messages to select the best clock to synchronize to and to determine the PTP state of each port, which is typically master, slave, or passive.

Only ordinary and boundary clocks participate in the self-organizing hierarchy using the BMCA. Transparent clocks are simply forwarding devices and do not serve or synchronize to the time. Fig. 3 shows an example hierarchy that may result from running the BMCA. In this example, Grandmaster Clock 2 determines that it is not the best clock on the network, so it stops transmitting *Announce* messages and goes into a passive state, effectively becoming a backup time server. Grandmaster Clock 2 continues to run the BMCA and will serve time when it determines that there is no longer a better clock on the network.

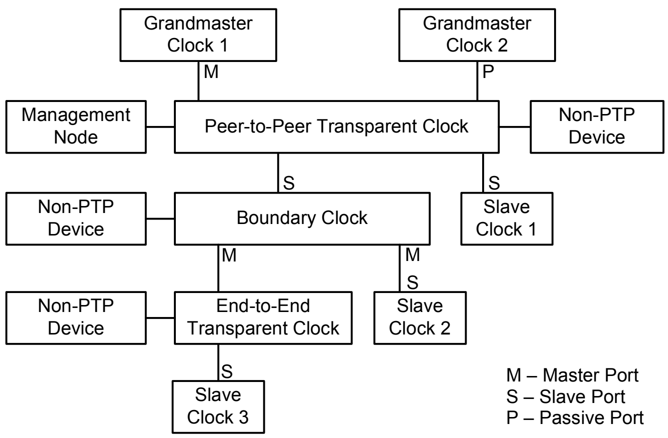

Fig. 3. Example PTP clock hierarchy

Fig. 4 and Fig. 5 show the message exchanges to synchronize slave clocks to master clocks. The communications path delay can be measured by using either the delay request-response mechanism or the peer delay mechanism. The delay request-response mechanism measures the end-to-end delay, while the peer delay mechanism measures the peer-to-peer delay, as shown in Fig. 1. Only one of the delay mechanisms can be used at a time in a PTP domain. The offset of the slave clock from the master clock is calculated by (1).

$$\text{offset} = (t_2 - t_1) - \text{path\_delay} \tag{1}$$

where:

$$\text{path\_delay} = \frac{t_{ms} + t_{sm}}{2} = \frac{(t_4 - t_3) + (t_2 - t_1)}{2}$$

If the peer delay mechanism is in use, the path delay (excluding the last link to the slave) is carried in the correction field of the *Sync* message. Fig. 4 and Fig. 5 show a one-step mode where the *Sync* and *Pdelay_Resp* messages carry t_1 and (t_2, t_3), respectively. To support hardware that may not be able to insert these time stamps as the messages are being transmitted, the protocol defines a two-step mode in which the *Sync* and *Pdelay_Resp* messages are followed by *Follow_Up* and *Pdelay_Resp_Follow_Up* messages carrying t_1 and t_3, respectively. While two-step *Sync* messages add traffic to the network, a two-step *Pdelay_Resp* adds traffic to only the link connecting the peers.

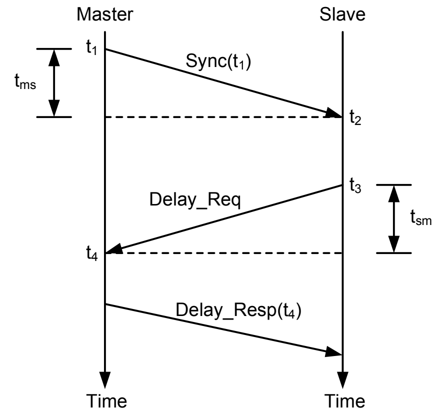

Fig. 4. Synchronization with delay request-response mechanism

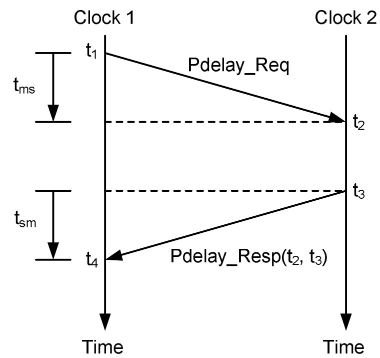

Fig. 5. Synchronization with peer delay mechanism

C. PTP Profiles

PTP defines a number of attribute options (e.g., transport over IEEE 802.3 Ethernet or UDP over IPv4) and optional features (e.g., unicast message negotiation). However, one of the goals of the PTP standard is to make it possible to set up and run a PTP network with minimal device settings and administration. The concept of PTP profiles allows organizations or industry groups to specify a subset of options and features as well as default values for protocol attributes that will meet the performance requirements of applications in the domain and eliminate or minimize device settings. PTP profiles can extend the protocol by defining an alternate BMCA, providing new transport mapping, and appending profile-specific tag, length, value (TLV) triplets to PTP messages. The most popular profiles are the default profiles, telecom profile (ITU-T G.8265.1), and power system profile (IEEE C37.238). The PTP standard defines the two default profiles, one for the delay request-response mechanism and the other for the peer delay mechanism. These two profiles specify the default values for the protocol attributes, but other attributes and features are still optional. The telecom profile is used to transfer frequency in a point-to-point fashion and is unlikely to be used in a power system environment. A future telecom profile (ITU-T G.8275.1) will distribute time. The power system profile (IEEE C37.238) was released in 2011 and is expected to be the dominant profile for power system applications in the near future. Table II summarizes the key profile provisions of IEEE C37.238.

TABLE II
IEEE C37.238 KEY PROFILE PROVISIONS

Profile Option	Value
BMCA	Default BMCA
Transport	IEEE 802.3/Ethernet with virtual local-area network (VLAN) tagging
Delay mechanism	Peer delay only
TLV triplets	*AlternateTimeOffsetIndicator* TLV, IEEE C37.238 TLV
Management	Simple Network Management Protocol (SNMP), support required of only grandmaster clocks

PTP distributes Coordinated Universal Time (UTC). The *AlternateTimeOffsetIndicator* TLV carries local time information for applications that need local time. The IEEE C37.238 TLV contains three pieces of information. The first piece of information is the *GrandmasterID*, which is a number intended to be used for *SmpSynch* in Sampled Value messages when the grandmaster clock is synchronized to some external time that is not a global reference. *SmpSynch* identifies the time source used in time-stamping the samples. The second piece of information is the *GrandmasterTimeInaccuracy*, which represents the accuracy with which the grandmaster clock is synchronized to external time (e.g., Global Positioning System [GPS] time). This is used by IEDs to indicate the time quality in IEC 61850 time stamps. The third piece of information is the *NetworkTimeInaccuracy*, which is the inaccuracy accumulated in the worst network path [4]. The use of *NetworkTimeInaccuracy* is undefined. IEEE C37.238 specifies that the inaccuracy introduced by a transparent clock shall not exceed 50 nanoseconds, which allows for a cascade of up to 16 transparent clocks for synchronization accuracies of less than 1 microsecond.

III. TEST NETWORKS AND RESULTS

To illustrate the effect of network elements and traffic on PTP synchronization accuracy, a number of simple networks were set up and the synchronization accuracy was measured using a PTP test device. A 1 pps signal from the slave device was fed into the test device, which served as a PTP master. To measure the synchronization accuracy, the test device compares this 1 pps signal with its own internally generated 1 pps signal. For each test, the synchronization interval was 1 second and the delay request-response (end-to-end) mechanism was used to measure the path delay between the slave and master. The delay interval was 1 second. To generate network traffic, two IEDs (referred to as Generic Object-Oriented Substation Event [GOOSE] devices) were configured to each publish eight GOOSE messages of varying sizes with each message triggering (i.e., data changing) every 8 milliseconds.

The first test network was a direct connection between the slave and the master, as shown in Fig. 6. In this setup, there was no other time-critical traffic through the communications channel. The time error distribution is shown in Fig. 7. It serves as the best-case result and as a basis to compare the results of the other tests.

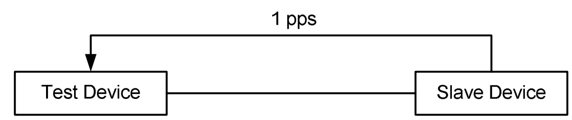

Fig. 6. Test 1: direct connection

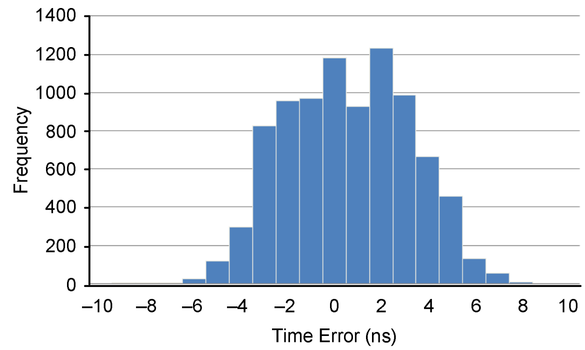

Fig. 7. Time error distribution for direct connection (Test 1)

The second test network connected the slave and master through a normal, nontransparent Ethernet switch, as shown in Fig. 8. There was no other network traffic besides the PTP messages. Fig. 9 shows the time error distribution for this test. The time error distribution shown in Fig. 10 was obtained by performing the test for a second time with the master and slave connections swapped on the switch. This is done by connecting the master device to the port that was previously connected to the slave and vice versa. These results show an offset of about 340 nanoseconds due to communications path asymmetry as well as a larger spread compared with the direct connection. If the offset is not manually compensated for, it can accumulate over a cascade of switches, as the next test network shows.

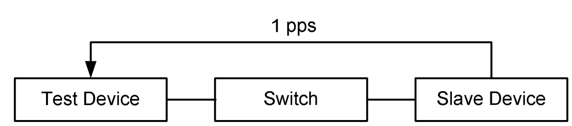

Fig. 8. Test 2: connection through a nontransparent switch

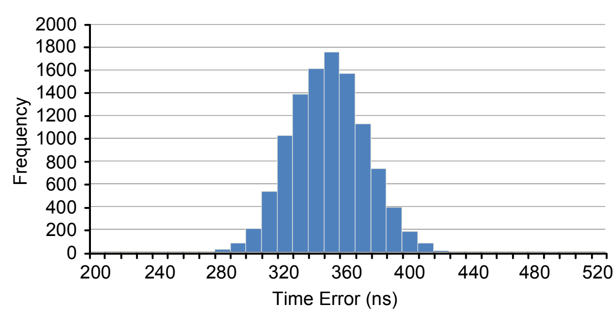

Fig. 9. Time error distributions for connection through a nontransparent switch (Test 2)

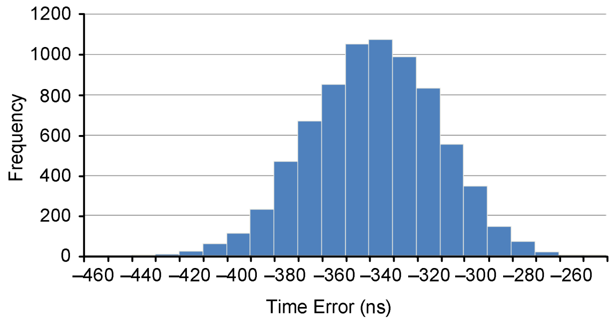

Fig. 10. Time error distributions for connection through a nontransparent switch with swapped connections (Test 2)

The third test network connected the master and slave through a cascade of three nontransparent switches, as shown in Fig. 11. The results in Fig. 12 show the accumulation of the errors introduced by the individual switches.

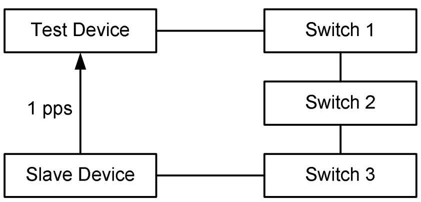

Fig. 11. Test 3: connection through a cascade of nontransparent switches

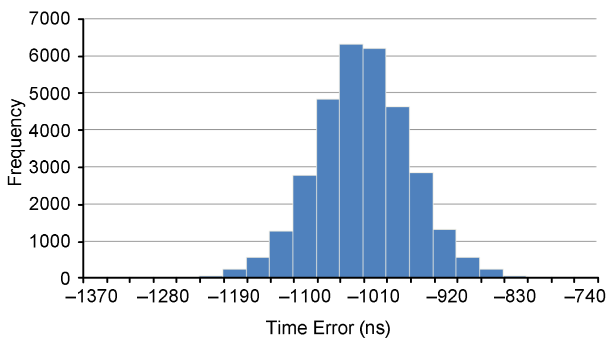

Fig. 12. Time error distribution for connection through a cascade of nontransparent switches (Test 3)

The setup for the fourth test network was the same as the third but with two GOOSE devices connected to Switches 1 and 3, as shown in Fig. 13. The time error distribution was similar to that of the third test network, as shown by the graph in Fig. 14, but there were occasional spikes in the time error, as shown by the graph in Fig. 15. The spikes are a result of packet delay variation (PDV) caused by increased queuing in the switch due to the large amount of traffic. Even though control algorithms, which are used to correct the slave clock to match that of the master, can be designed to reject these spikes, there is usually a tradeoff between PDV rejection and the response to actual network changes.

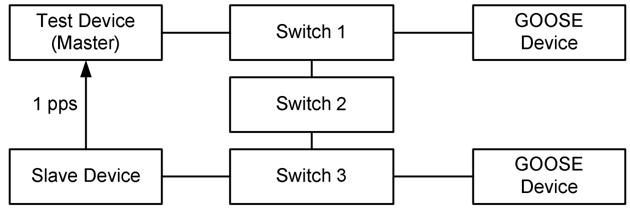

Fig. 13. Test 4: connection through a cascade of nontransparent switches with two GOOSE devices

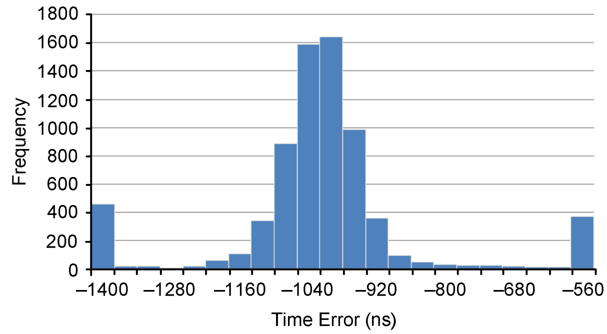

Fig. 14. Time error distribution with a cascade of three nontransparent switches and two GOOSE devices (Test 4)

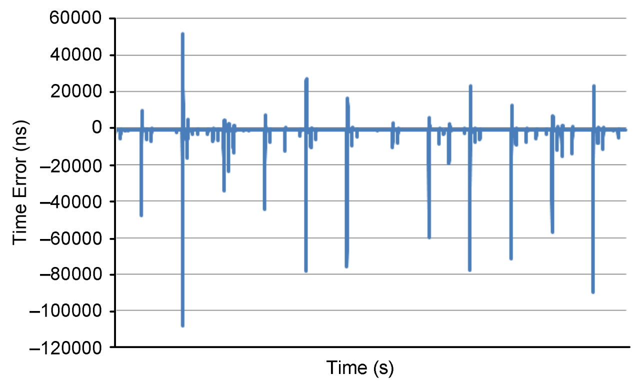

Fig. 15. Variance of time error over time (Test 4)

The fifth test network (shown in Fig. 16) was set up like Fig. 8 but with the nontransparent switch replaced by a transparent switch. The results in Fig. 17 and Fig. 18 show a better accuracy with much smaller asymmetry. The distribution in Fig. 18 was obtained by swapping the connections on the switch between the master and the slave devices and repeating the test.

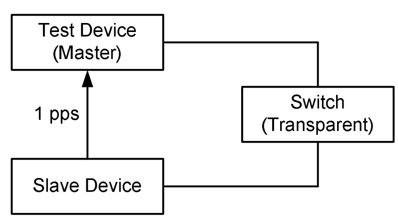

Fig. 16. Test 5: connection through a transparent switch

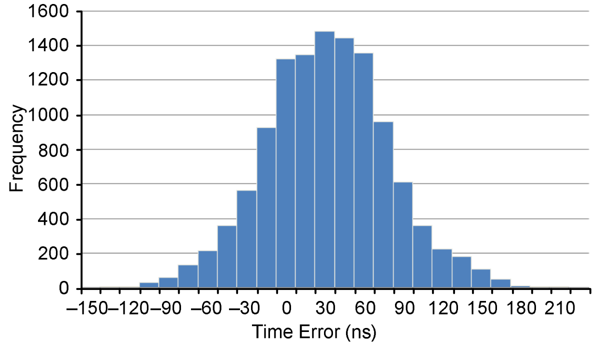

Fig. 17. Time error distribution for connection through a transparent switch (Test 5)

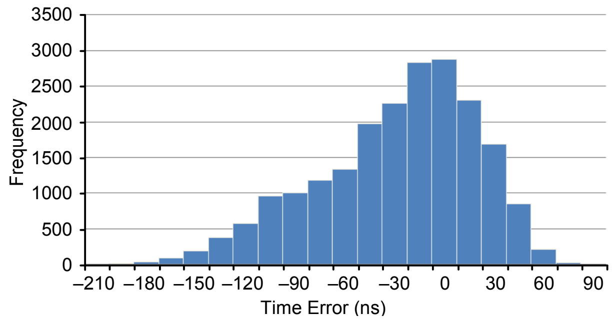

Fig. 18. Time error distribution for connection through a transparent switch with swapped connections (Test 5)

The sixth and last test network (shown in Fig. 19) was similar to the fifth but with network traffic generated by two GOOSE devices. Unlike in the fourth network, the traffic had an insignificant effect on the accuracy, as shown in Fig. 20. Because the transparent switch accounts for the residence time, the length of time that PTP messages spend in the switch has almost no effect on accuracy. Longer residence times may have a noticeable effect on accuracy if the frequency of the free-running internal oscillator used to measure the residence time is off from nominal.

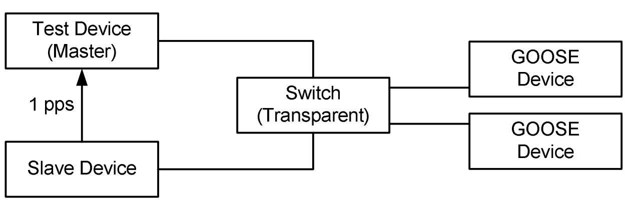

Fig. 19. Test 6: connection through a transparent switch with two GOOSE devices

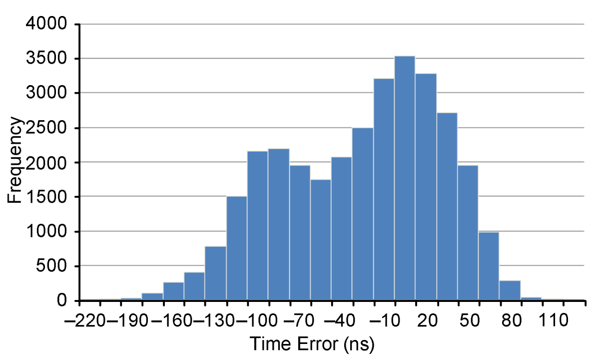

Fig. 20. Time error distribution with a transparent switch and two GOOSE devices (Test 6)

IV. PTP Network Design Considerations

PTP enables, but does not guarantee, submicrosecond accuracy and precision. Accuracy refers to the mean of the time error measurements between the master and slave clocks. Precision is a measure of the deviation of the error measurements from the mean, usually represented by the standard deviation. The achievable synchronization accuracy and precision depend on device-specific characteristics and network design.

The following device-specific characteristics affect accuracy and precision:

- The resolution of the internal PTP clock. For example, if the PTP clock is advanced or updated every 8 nanoseconds (a tick generated by a 125 MHz oscillator), the accuracy and precision cannot be better than 8 nanoseconds.
- The stability of the oscillator that drives the PTP clock. If the oscillator is unstable, the real tick rate or period between clock pulses could significantly vary over a synchronization interval before the next update.
- The accuracy of event message time stamps. Because of nondeterminism in task switching and scheduling, software-based time-stamping is usually only accurate to a few milliseconds. Hardware-assisted time-stamping is required for submicrosecond accuracy and precision.
- The noise of the control loop used to correct the slave clock to match the time of, and to tick at the same rate as, the master clock. The control (or servo) algorithm is often unable to adjust the slave clock to tick at exactly the same rate as the master.

The accuracy and precision specifications of the device include the combined effect of these factors. Other than selecting the appropriate device, the described factors are typically beyond the control of the network designer. However, the following network-dependent factors are within the control of the network designer:

- The delay variations that PTP event messages experience in the network. This affects synchronization precision. As the test results show, PTP can be made insensitive to network delay variations by using transparent and boundary clocks. The results in Fig. 14 show that large PDVs are likely in modern power system networks because of the increasing amount of real-time, protection-related traffic they handle (e.g., Sampled Values and GOOSE). Therefore, for critical applications requiring highly precise time, it is recommended to use transparent clocks, especially for new networks.
- Whether or not the communications path delay is measured and compensated for. Synchronization accuracy is affected if the path delay is not accounted for. The end-to-end or peer-to-peer delay measurement mechanisms provided by the protocol can be used to automatically measure and compensate for the network path delay. The communications path delay can be significant in networks where the PTP

messages traverse multiple switches before reaching their destination. It is therefore recommended to use one of the path delay mechanisms to measure and compensate for communications path delays.

- The asymmetry in the network. PTP assumes a symmetrical communications path (i.e., it takes the same time to send a message from Device A to Device B as it takes to send one from Device B to Device A). If this does not hold true for a network, the synchronization accuracy is affected. The protocol does not provide a means to measure network asymmetry, so asymmetry may therefore have to be manually measured and compensated for. This can be done by characterizing the network by understanding the traffic and compensating for the inaccuracies due to asymmetries.

It is worth noting that unlike some other power system protocols, PTP is able to withstand some message losses.

V. Conclusion

The need for synchronized time for power systems began with the requirement to time-stamp events for postmortem sequence-of-event analysis. An accuracy of 1 millisecond was adequate for such applications. Modern power system applications (such as synchrophasors, Sampled Values, and traveling wave fault location) require submicrosecond accuracies. These stricter accuracy requirements, together with a growing need to provide time synchronization via Ethernet, have made PTP an attractive (and perhaps the only) option to satisfy both requirements. However, migrating from traditional IRIG-B to PTP will likely involve making decisions that are a tradeoff between costs and synchronization accuracy.

VI. References

[1] D. Dolezilek, "Ethernet Design for Teleprotection and Automation Requires a Return to First Principles to Improve First Response," proceedings of the 14th Annual Western Power Delivery Automation Conference, Spokane, WA, March 2012.

[2] E. O. Schweitzer, III, D. Whitehead, S. Achanta, and V. Skendzic, "Implementing Robust Time Solutions for Modern Power Systems," proceedings of the 14th Annual Western Power Delivery Automation Conference, Spokane, WA, March 2012.

[3] IEEE Standard 1588-2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.

[4] IEEE Standard C37.238-2011, IEEE Standard Profile for Use of IEEE 1588™ Precision Time Protocol in Power System Applications.

VII. Biographies

Steve T. Watt received his B.S. in mechanical engineering from Virginia Polytechnic Institute and State University. He worked in the information technology industry for over 20 years at Hewlett Packard before joining Schweitzer Engineering Laboratories, Inc. (SEL) in 2012. Steve is currently the product manager for precision time products in the time and communications group at SEL.

Shankar Achanta received his M.S. in electrical engineering from Arizona State University in 2002. He joined Schweitzer Engineering Laboratories, Inc. (SEL) in 2002 as a hardware engineer, developing electronics for communications devices, data acquisition circuits, and switch mode power supplies. Shankar received a patent for a self-calibrating time code generator while working at SEL, and he is an inventor on several patents that are pending in the field of precise timing and wireless communication. He currently holds the position of research and development manager for the precise time and wireless communications group at SEL.

Hamza Abubakari received his M.S. in electrical engineering from The University of Akron in 2008. He joined Schweitzer Engineering Laboratories, Inc. (SEL) in 2008 as a software engineer, focusing on IEC 61850. He was the lead engineer in developing the IEC 61850 communication stack for SEL relays and other devices. He also led a team to develop a PTP firmware component for all SEL devices.

Eric Sagen received his B.S. in electrical engineering from Washington State University in 1997. He then joined General Electric in Pennsylvania as a product engineer. In 1999, Eric joined Schweitzer Engineering Laboratories, Inc. as a distribution product engineer. Shortly after, he was promoted to lead distribution product engineer. Eric transferred to the time and communications group in 2006 and is currently a lead product engineer. He is certified in Washington as an Engineer in Training (EIT).