

White Paper

Präzise Uhrzeitsynchronisation

Der Standard IEEE 1588

Inhaltsverzeichnis

Präzise Uhrzeitsynchronisation Der Standard IEEE 1588

1	Einleitung	3
2	Wozu Zeitsynchronisation	3
3	Bisherige Lösungen	4
4	PTP Anwendungen	4
5	Die Funktionsweise von PTP	6
6	Die Implementierung von PTP	7
7	Ergebnisse	8
8	Kooperation zwischen Hirschmann und der Zürcher Hochschule Winterthur	9
9	Weitere Entwicklungen	9
10	Zusammenfassung	10
	Anhang 1: Das PTP Protokoll im Detail	11
	Anhang 2 – Die Implementierung von PTP im Detail	14

Präzise Uhrzeitsynchronisation

Der Standard IEEE 1588

1 Einleitung

Präzise Zeitinformation ist insbesondere für verteilte Systeme in der Automatisierungstechnik wichtig. Mit dem in IEEE 1588 definierten Precision Time Protocol (PTP) ist es erstmals möglich, über Ethernet-Netzwerke verteilte Uhren auf weniger als eine Mikrosekunde Genauigkeit zu synchronisieren. Dabei sind die Ansprüche an die lokalen Uhren sowie an die Netz- und Rechenkapazität vergleichsweise gering. Dieses White Paper gibt einen Überblick über Anwendungsmöglichkeiten und Funktion des Precision Time Protocols. Im Anhang sind zusätzlich technisch weitergehende Informationen zu finden, die Details zur Funktionsweise erläutern, sowie Hinweise zur Umsetzung von PTP in Geräte geben.

2 Wozu Zeitsynchronisation

Wir verwenden Uhren, um uns mit Personen oder Vorgängen zu synchronisieren. Die notwendige Genauigkeit der Uhr hängt dabei von der Anwendung ab. Wer einen Zug erreichen will, sollte die Uhrzeit auf eine Minute genau im Blick haben. Beim sportlichen Wettkampf können hundertstel Sekunden den Ausschlag geben, und Antriebe innerhalb einer Verpackungsmaschine benötigen eine Synchronisation im Bereich von Mikrosekunden.

Viele technische Systeme verfügen über einen Zeitbegriff. Eine implizite Systemzeit liegt vor, wenn keine eigentliche Uhr vorhanden ist und das zeitliche Verhalten durch Abläufe in der Hard- und Software bestimmt wird. Dies ist in vielen Systemen oft ausreichend. Realisiert wird eine implizite Systemzeit beispielsweise über regelmäßige Triggerereignisse an jeden Teilnehmer, die den Beginn einer Zeiteinheit kennzeichnen und dann die entsprechende Aktionen auslösen.

Die Systemzeit liegt explizit vor, wenn sie durch eine Uhr repräsentiert wird. Insbesondere in komplexen Systemen ist dies oft notwendig. Damit wird die Kommunikation und die Ausführung voneinander entkoppelt. Nicht jede Uhr geht allerdings genau. Ab und zu muss überprüft werden, ob die Abweichung im Rahmen liegt und die Uhr bei Bedarf korrigiert werden muss. Dazu ist eine Kommunikation zwischen den einzelnen Uhren erforderlich.

Beim Stellen bzw. Synchronisieren von Uhren sind zwei Effekte zu beachten: Zum einen laufen unabhängige Uhren zunächst zeitversetzt. Um sie zu synchronisieren, wird die ungenauere Uhr auf die Zeit der genaueren Uhr eingestellt (Offset Korrektur). Weiterhin laufen reale Uhren auch nicht exakt gleich schnell. Daher ist eine ständige Regelung der Laufgeschwindigkeit der ungenaueren Uhr notwendig (Drift Korrektur).

3 Bisherige Lösungen

Es gibt bisher verschiedene Methoden, um verteilte Uhren über ein Netzwerk zu synchronisieren. Die gebräuchlichsten sind das Network Time Protocol (NTP) und das daraus abgeleitete einfachere Simple Network Time Protocol (SNTP). Diese Verfahren sind in LANs (Local Area Networks) oder im Internet weit verbreitet und erlauben Genauigkeiten bis in den Millisekunden - Bereich. Eine weitere Möglichkeit ist die Verwendung von Funksignalen der GPS-Satelliten. Dies bedingt jedoch relativ teure GPS-Empfänger bei jeder Uhr sowie entsprechende Antennen. Damit lassen sich zwar hochpräzise Uhren realisieren, aus Kosten- und Aufwandsgründen ist dies jedoch vielfach nicht möglich.

Eine weitere Lösung besteht darin, einen hochgenauen Zeittakt (z.B. Pulse per Second Signal) über separate Leitungen an jeden Teilnehmer zu schicken. Dies erfordert jedoch einen beträchtlichen zusätzlichen Verkabelungsaufwand.

Hier setzt das in IEEE 1588 beschriebene Precision Time Protocol (PTP) an. Es wurde mit den folgenden Zielen entwickelt:

- Synchronisationsgenauigkeit im Sub-Mikrosekunden-Bereich
- Minimale Anforderungen an die Prozessorleistung und Netzwerkbandbreite, wodurch die Implementierung auch auf einfachen und preisgünstigen Geräten möglich ist
- Geringer Administrationsaufwand
- Verwendung über Ethernet-Netzwerke, aber auch über andere multicastfähige Netzwerke
- Spezifikation als internationaler Standard

4 PTP Anwendungen

Die Idee zu PTP entstand Ende der 90er Jahre in den USA bei Agilent Technologies im Bereich Messtechnik. Das dort entwickelte Verfahrensprinzip wurde als Vorschlag bei IEEE eingereicht und als Standard IEEE 1588 ausgearbeitet. Ende 2002 wurde PTP unter der Bezeichnung "1588TM - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems"

verabschiedet. Zusätzlich wurde PTP im Mai 2004 auch als IEC Standard übernommen und ist dort unter der Bezeichnung IEC 61588 erschienen.

PTP findet zunehmend Interesse in unterschiedlichsten Anwendungen. In der Automatisierungstechnik ist PTP überall dort gefragt, wo Vorgänge präzise synchronisiert werden müssen. Hier ist im weitesten Sinne der Bereich Motion Control ein wichtiges Einsatzfeld. Denn PTP hilft, Antriebe beispielsweise innerhalb eines Roboters oder einer Druck-, Verpackungs- oder Papierverarbeitungsmaschine zu synchronisieren. Kooperierende Roboter werden untereinander über hochpräzise Uhren gekoppelt, oder komplette Maschinen- oder Anlagenteile werden eng über PTP verbunden, so dass die ablaufenden Prozesse zeitlich exakt aufeinander abgestimmt sind. Synchron laufende Uhren in jeder Komponente ermöglichen es, dezentrale Strukturen aufzubauen und die Ausführung der Prozesse von der Kommunikation und Verarbeitung der Steuerbefehle zu entkoppeln.

Aus diesem Grund ist Zeitsynchronisation nach IEEE 1588 mittlerweile Bestandteil fast aller zukünftiger Echtzeit-Automatisierungsprotokolle. CIPsync, Teil des Ethernet/IP Frameworks der ODVA, setzt für Motion Control-Anwendungen voll auf PTP. PROFINet (PNO) hat PTP auf seiner Roadmap als Synchronisationsprotokoll stehen, und ETHERNET Powerlink (EPKG) wird PTP ebenfalls zur Synchronisation von Echtzeitsegmenten nutzen.

Aber auch außerhalb der Automatisierungstechnik arbeiten viele Firmen an der Evaluierung und Implementierung von PTP. Überall, wo Messwerte erfasst und in Beziehung zueinander gebracht werden müssen, ist PTP eine gefragte Lösung. In Energieverteilanlagen werden Kenngrößen wie Ströme und Spannungen in verteilten Sensoren gemessen, an zentraler Stelle miteinander verknüpft und ausgewertet. Turbinensteuerungen bedienen sich des PTP-Protokolls, um noch effizientere Anlagen aufzubauen. Und zur Überwachung von Prozessen werden dezentral erfasste Ereignisse mit präzisen Zeitstempeln markiert und zur Protokollierung bzw. Analyse an die Leitstelle übermittelt.

In der Messtechnik wird PTP eingesetzt zur Korrelation von dezentral erfassten physikalischen Größen, etwa in der Hochfrequenz-Messtechnik. Geowissenschaftler verwenden PTP, um seismische Messgeräte über große Entfernungen zu synchronisieren und damit Erdbebenherde exakter lokalisieren zu können. In der Telekommunikation wird daran gedacht, Netzwerke über PTP zu synchronisieren oder Mobilfunk-Basisstationen mit präzisen Zeittakten zu versorgen. Darüber hinaus besteht Interesse an Zeitsynchronisation nach IEEE 1588 in Bereichen wie Sicherheitstechnik, Automobiltechnik oder in militärischen Anwendungen.

Alle diese Applikationen sind momentan in Entwicklung oder im Prototypenstadium. In den nächsten Jahren werden jedoch viele dieser Anwendungen in regulären Produkten auf den Markt kommen, und PTP eine weite Verbreitung finden.

5 Die Funktionsweise von PTP

PTP kennt zwei Arten von Uhren, Master und Slaves. Eine Uhr in einem Endgerät wird als Ordinary Clock bezeichnet, eine Uhr in einer Übertragungskomponente wie einem Ethernet Switch als Boundary Clock. Ein Master, der idealerweise von einer Funkuhr oder einem GPS Empfänger gesteuert wird, synchronisiert dabei jeweils die an ihn angeschlossenen Slaves.

Der Synchronisationsvorgang wird in zwei Phasen gegliedert. Zuerst wird eine Korrektur der zeitlichen Differenz zwischen Master und Slave durchgeführt, die Offset Korrektur. Dazu sendet der Master zyklisch eine Synchronisationsnachricht – SYNC message – mit einem geschätzten Wert der Uhrzeit zu den angeschlossenen Slaves. Parallel wird der Zeitpunkt, an dem die Nachricht den Sender verlässt, so präzise wie möglich gemessen, möglichst per Hardwareunterstützung direkt auf dem Medium. Der Master sendet anschließend in einer zweiten Nachricht – FollowUp message – diesen tatsächlichen exakten Sendezeitpunkt der korrespondierenden Sync Nachricht an die Slaves. Diese messen ebenfalls möglichst exakt den Empfangszeitpunkt dieser Nachrichten, und können daraus den Korrekturwert (Offset) zum Master berechnen. Die Slave Uhr wird anschließend um diesen Offset korrigiert. Hätte die Übertragungsstrecke keine Verzögerung, dann wären nun beide Uhren synchron.

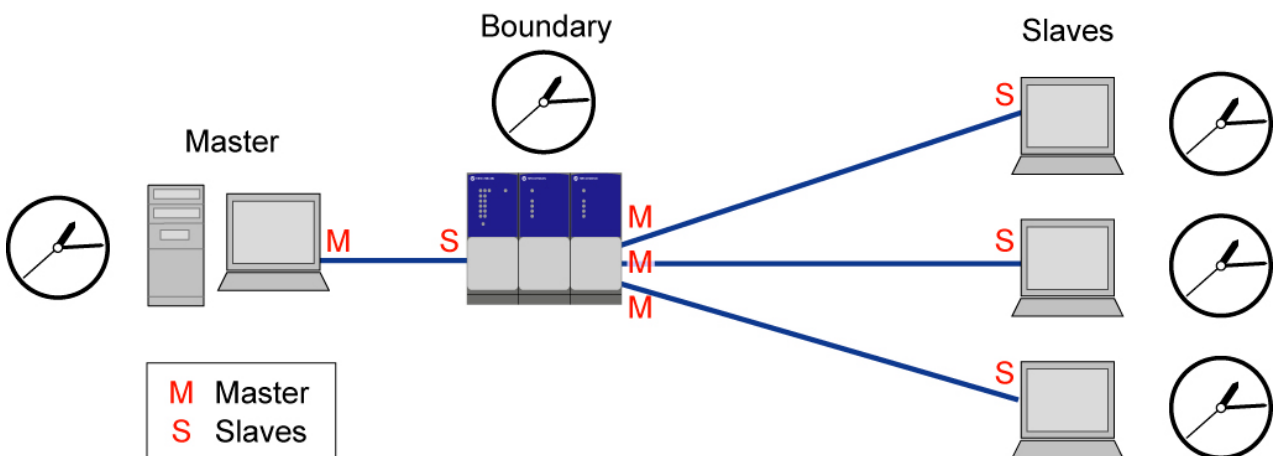


Abbildung 1: Boundary Clock Switches arbeiten gegenüber der Master Clock als Slave, und versorgen als Master die weiteren angeschlossenen Slaves.

Die zweite Phase der Synchronisation, die Delay- Messung, bestimmt die Laufzeit zwischen Slave und Master. Diese wird mittels sogenannter Delay Request und Delay Response Nachrichten auf ähnliche Weise bestimmt, und die Uhren entsprechend justiert.

Entscheidend für eine Delay-Messung und deren Genauigkeit ist jedoch eine symmetrische Verzögerung zwischen Master und Slave, d.h. gleiche Werte für Hin- und Rückrichtung. Bei direkter Verbindung über ein Stück Kabel ist dies praktisch erfüllt. Befinden sich jedoch Netzwerkkomponenten wie Switches oder Router zwischen den Teilnehmern, ist dies nicht mehr gegeben. Aus diesem Grund wurde für solche

Geräte die sogenannte Boundary Clock definiert. Ein solcher Switch enthält eine Uhr, die sich als Slave Clock auf einen angeschlossenen Master synchronisiert, und sich dann wiederum an jedem weiteren Port als Master verhält und die daran angeschlossenen Slaves synchronisiert. Dadurch geschieht die Synchronisation immer in einer Punkt zu Punkt Verbindung, die sehr symmetrische Laufzeiten und praktisch keinen Jitter aufweist. Wartezeiten oder Jitter bei dem Durchlauf durch die Switches haben damit keinen Einfluss auf die Genauigkeit der Synchronisation.

Eine detaillierte Beschreibung der Funktionsweise findet sich in Anhang 1.

6 Die Implementierung von PTP

Soll in einem System das Precision Time Protokoll genutzt werden, so ist der PTP- Protokollstack zu implementieren. Dieser ist sehr kompakt realisierbar und hat nur minimale Anforderungen an die Prozessorleistung und die Netzwerkbandbreite. Dies ist sehr wichtig bei der Implementierung in einfachen und preisgünstigen Geräten. PTP lässt sich selbst in Embedded Systemen mit einfachen 16 oder 32 Bit Mikrocontrollern problemlos integrieren. Einzige Anforderung zur Erreichung einer hohen Präzision ist eine möglichst exakte Erfassung der Sende- und Empfangszeiten der PTP-Messages. Diese muss sehr hardwarenah erfolgen (z.B. direkt in der Treiber-Software) und mit möglichst hoher Genauigkeit. Damit beschränkt bei Implementierungen als reine Softwarelösung die Architektur und Performance des Systems unmittelbar die erreichbare Genauigkeit.

Bei Einsatz von zusätzlicher Hardwareunterstützung zur Zeitstempelung kann die Präzision erheblich gesteigert und praktisch unabhängig von der Software gestaltet werden. Dazu ist eine kleine Logik notwendig, die beispielsweise als FPGA realisiert oder in ASICs direkt am Netzwerkzugang integriert werden kann.

Wird PTP über Ethernet-Netzwerke eingesetzt, so ist besonderer Augenmerk auf die Netzwerk-Infrastruktur zu legen. Während Hubs aufgrund ihrer praktisch konstanten Durchlaufzeit die Genauigkeit des Protokolls nahezu nicht beeinflussen, sind beim Einsatz von Switches, wie in Kapitel 5 bereits beschrieben, die Laufzeiten zu betrachten. PTP misst zwar die Laufzeiten im Netzwerk und korrigiert die Uhren entsprechend. Laufzeitvarianzen, wie sie bei Switches jederzeit vorkommen können, führen jedoch zu Ungenauigkeiten.

Da Switches die empfangenen Datenpakete komplett speichern, und durch Warteschlangeneffekte die Aussendung unter Umständen erst deutlich verzögert erfolgen kann, können hier erhebliche Schwankungen auftreten. Bei geringer Netzlast wirkt sich dieser Effekt kaum aus, bei größerer Netzlast oder auch nur kurzfristigen Lastsituationen kann dies jedoch die Synchronisationsgenauigkeit deutlich verschlechtern.

Dem begegnet man mit dem Einsatz von so genannten Boundary-Clock-Switches. Diese enthalten eine eigene PTP-Instanz, die gegenüber der angeschlossenen Master-Uhr als PTP-Slave arbeitet und damit von ihm exakt synchronisiert wird. Gegenüber den angeschlossenen Endgeräten, den PTP-Slaves, arbeitet dann jeder einzelne Switch-Port wiederum als PTP-Master und synchronisiert diese Slaves mit seiner internen Uhrzeit. Damit sind jegliche Laufzeitschwankungen und Wartezeiten in den Switches kompensiert, auch über größere Ethernet-Netze hinweg lassen sich so höchste Genauigkeiten erreichen.

7 Ergebnisse

Hirschmann gehört zu den ersten Herstellern, die das Precision Time Protocol implementiert und optimiert haben. Dazu wurde ein Softwarestack entwickelt, der das Protokoll sehr effizient realisiert sowie ein Chip, welcher das Verfahren für höchste Präzision unterstützt.

Die Implementierung erreicht als reine Software-Lösung Genauigkeiten von etwa 5 bis 50us. Mit Hardwareunterstützung für die Zeitstempelerfassung ist die Präzision nochmals deutlich höher.

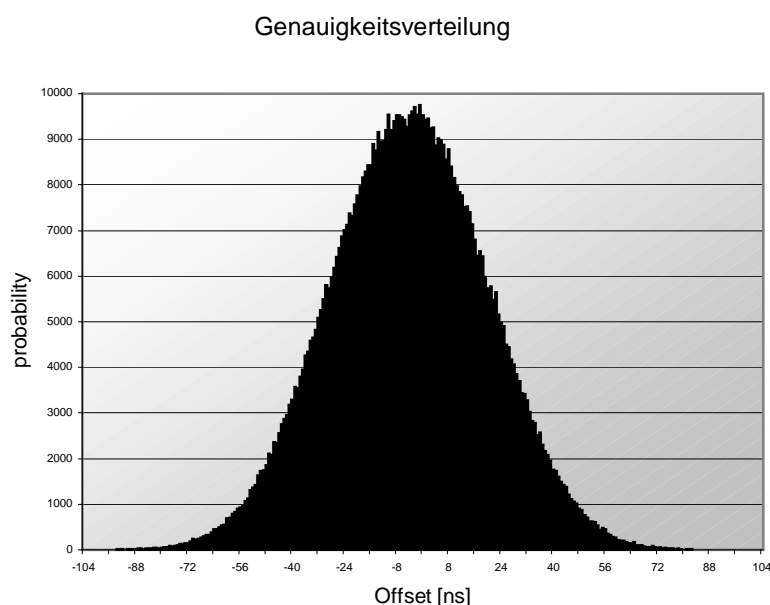


Abbildung 2: Das Precision Time Protocol erreicht mit Hardwareunterstützung Genauigkeiten von unter 100ns

In einer Pilotanlage, bei der mehrere Ordinary-Clocks über einen Ethernet-Switch mit Boundary-Clock-Funktion verbunden wurden, konnte eine typische Genauigkeit von +/-60ns erreicht werden, praktisch unabhängig von der Netzlast oder der Auslastung der CPU. Für die Industrial-Ethernet-Switches der MICE-Familie von Hirschmann sind Varianten mit PTP-Boundary-Clock verfügbar.

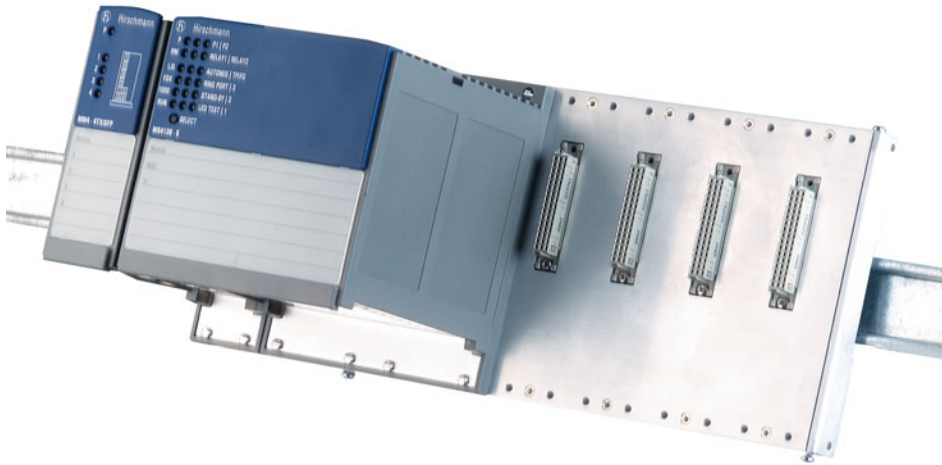


Abbildung 3: Die Industrial Ethernet Switches der MICE Serie unterstützen das Precision Time Protocol nach IEEE 1588

8 Kooperation zwischen Hirschmann und der Zürcher Hochschule Winterthur

In Kooperation mit der Zürcher Hochschule Winterthur (ZH) arbeitet Hirschmann seit einiger Zeit an der Weiterentwicklung und Optimierung von IEEE 1588. Im Rahmen dieser Kooperation bietet die ZH interessierten Anwendern Unterstützung bei der Realisierung von PTP-Ordinary-Clocks an. Dies umfasst Software-Stacks und VHDL-Designs, die auf der von Hirschmann entwickelten Technologie basieren, sowie zusätzliche Dienstleistungen wie Evaluierungs-Kits, Training, oder Designunterstützung. Nähere Informationen hierzu sind auf der Webseite „<http://ines.zhwin.ch/ieee1588/index.html>“ zu finden.

9 Weitere Entwicklungen

IEEE 1588 ist in der Automatisierungs- und Messtechnik breit akzeptiert. Doch auch für andere Branchen ist das Potenzial des Verfahrens interessant. Um zusätzliche Anforderungen zu erfüllen wurde im Februar 2005 im IEEE Gremium das Projekt P1588 gestartet, das den Standard in folgende Richtungen erweitern soll:

- **Höhere Genauigkeit:** Ziel ist die Erreichung von Genauigkeiten im Nanosekunden Bereich und darunter. Voraussetzung dafür sind hochwertigere Oszillatoren, kürzere Sync-Intervalle, höhere Auflösung der Zeitstempel und bessere Symmetrie.
- **Ausfallsicherheit:** Bei Ausfall der Master-Uhr muss die Rekonfiguration so erfolgen, dass die Qualität der Synchronisation nicht beeinträchtigt wird.
- **SNMP-Management:** In Ethernet Netzwerken ist SNMP das am weitesten verbreitete Netzmanagement Protokoll. Deshalb soll eine PTP MIB spezifiziert werden.

- **Verschiedenartige Ethernet Header:** Die Berücksichtigung von Tagged Ethernet Frames (VLAN), IPv6 Paketen und Felder für Protokollerweiterungen sollen ermöglicht werden.
- **Transparent Clock:** In der Automatisierungstechnik werden oft Topologien mit Linien- und Ringstruktur eingesetzt. Die Aneinanderreihung von Boundary Clocks, von denen jede einen eigenen Regelkreis darstellt, ist darum problematisch, weil sich Fehler kumulieren. Als Alternative ist die Transparent Clock vorgeschlagen. Diese realisiert eine Ende-zu-Ende-Regelung (im Gegensatz zur Kaskade einzelner Regler). Die Switches messen die Durchlaufzeit der PTP-Meldungen und manipulieren die in den Meldungen enthaltenen Zeitstempel entsprechend. Mit Hilfe der Transparent Clock wird auch die schnelle Rekonfiguration des Netzwerks im Fehlerfall möglich.
- **Anwendung in anderen Netzwerktypen:** Dazu müssen einige Details wie Meldungsformat, Adressierung und Bezugspunkt der Zeitstempel spezifiziert werden. Kandidaten sind WLAN, DeviceNet und Telekom-Netzwerke.
- **Sicherheit:** In offenen Netzwerken muss der Synchronisationsmechanismus vor unerwünschtem Zugriff geschützt werden.

10 Zusammenfassung

PTP hat mittlerweile seine Tauglichkeit in den Labors bewiesen und ist nun auf dem Sprung in praktische Anwendungen. Man kann sicher sein, dass PTP in den nächsten Jahren eine weite Verbreitung finden wird und dass viele Anwendungen damit wesentlich effizienter und einfacher zu realisieren sein werden als mit bestehender Technologie.

Weitere Informationen unter:

http://www.hirschmann.de	Hirschmann Automation and Control GmbH
http://ieee1588.nist.gov/	IEEE 1588 Standardisierungsgruppe
http://ines.zhwin.ch/ieee1588/	Zürcher Hochschule Winterthur, PTP Stacks und Dienstleistungen

Anhang 1: Das PTP Protokoll im Detail

Der Standard IEEE 1588 beschreibt das Precision Time Protocol (PTP) inklusive der zugrundeliegenden Synchronisationsalgorithmen. Zusätzlich wird eine Initialisierungsphase beschrieben, die mittels des sogenannten Best Master Clock Algorithmus die genaueste Uhr im Netzwerk als die bestimmende Systemuhr auswählt.

Das Verhalten der Uhren wird im Standard über Zustandsmaschinen beschrieben, weiterhin werden Datentypen und deren Repräsentation definiert. Zur Verwaltung und Konfiguration der Uhren im Netzwerk definiert der Standard ein eigenes einfaches Managementprotokoll über das der Benutzer zum einen die Zustände aller beteiligten Uhren abfragen kann, aber auch die Konfigurationen vornehmen kann. So kann er z.B. bestimmen welche Uhr denn bevorzugt die Funktion des Grandmaster übernehmen soll.

Nicht spezifiziert sind die Verfahren zur Regelung der Uhren und die Anwenderschnittstelle (API), über die die Uhr von der Applikation angesprochen werden kann.

PTP basiert auf IP Multicast Kommunikation und ist nicht auf Ethernet beschränkt, sondern kann prinzipiell auf jedem multicastfähigen Bussystem eingesetzt werden. Die Multicast Kommunikation bietet den Vorteil der Einfachheit, es muß keine IP Adreßverwaltung auf den PTP Knoten implementiert werden. Speziell durch das verwenden von IP Multicast ist PTP für eine hohe Anzahl an PTP Knoten skalierbar.

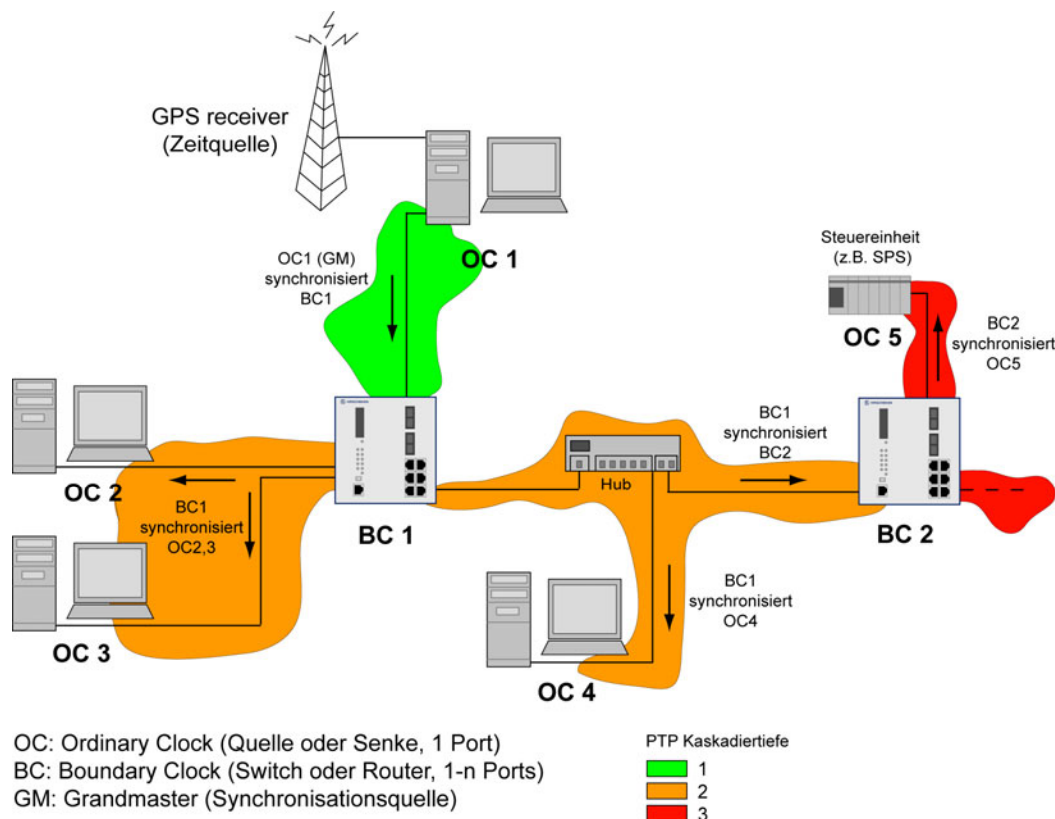


Abbildung 4: PTP Synchronisationsdomain

Zeitsynchronisation, der Algorithmus

Zur Zeitsynchronisation wird bei PTP die Uhrzeit des Masters so präzise wie möglich dem Slave mitgeteilt. Die Besonderheit besteht darin, sämtliche Verarbeitungszeiten und Laufzeiten durch das Verfahren so zu kompensieren, dass eine möglichst optimale Synchronisation hergestellt werden kann. Dazu gliedert sich der Synchronisationsvorgang in zwei Phasen.

Zuerst korrigieren Master und Slave ihre zeitlichen Differenz mit der Offset Messung. Bei dieser Offsetkorrektur sendet der Master zyklisch in definierten Intervallen (in der Grundeinstellung alle 2 Sekunden) eine eindeutige Synchronisationsnachricht – die SYNC Message – zum angeschlossenen Slave. Diese Sync- Nachricht beinhaltet die aktuelle Zeit der Master-Uhr. Da aber das Auslesen der Uhr, die Verarbeitung des Protokolls, das Durchlaufen des Kommunikationsstacks, und das Aussenden der Daten über den Ethernet-Controller eine undefinierte Zeit benötigt, ist die Zeitinformation in der Sync Message beim Verlassen des Masters bereits veraltet. Deshalb wird so nah wie möglich an der physikalischen Schnittstelle (idealerweise direkt am Ethernetport per Hardware) der tatsächliche Sendezeitpunkt der Message gemessen, und diese Information mittels einer zweiten Nachricht, der FollowUp Message, an den Slave gesendet.

Mit dem Empfang der FollowUp Nachricht berechnet der Slave unter Berücksichtigung des Empfangszeitstempel der Sync Nachricht, den Korrekturwert (Offset Θ) zum Master. Die Slave Uhr muss anschließend um diesen Offset korrigiert werden. Diese präzisen Zeitstempel ermöglichen es nun, daß zeitliche Schwankungen des darüberliegenden Protokoll Stacks – Protokoll Jitter – keine Bedeutung mehr haben.

Hätte die Übertragungsstrecke keine Verzögerung, dann wären nun beide Uhren bereits synchron.

Die zweite Phase des Synchronisationsvorgang bestimmt die Verzögerungszeit – (Delay, die Laufzeit auf dem Netzwerk) – zwischen Slave und Master, die Delay Messung. Hierzu sendet der Slave ein sogenanntes „Delay Request“ Paket an den Master und ermittelt dabei wieder die exakte Sendezeit der Nachricht. Der Master generiert einen Zeitstempel bei Empfang des Pakets und sendet die Empfangszeit in einem „Delay Response“ Paket an den Slave zurück. Aus dem lokalen Sende- und dem Empfangszeitstempel des Masters, ermittelt der Slave die Verzögerungszeit zwischen Slave und Master. Die Delay Messung erfolgt unregelmäßig und in größeren Zeitintervallen (4 - 60 Sekunden) als eine Offset Messung. Dadurch wird das Netz und insbesondere die Endgeräte nicht zu sehr belastet. Entscheidend für eine Delay Messung und dessen Genauigkeit ist jedoch eine symmetrische Verzögerung ($T_{\text{Delay_Hin}} = T_{\text{Delay_Rück}} = T_{\text{Delay}}$) zwischen Master und Slave.

Abbildung 5 illustriert den Synchronisationsalgorithmus.

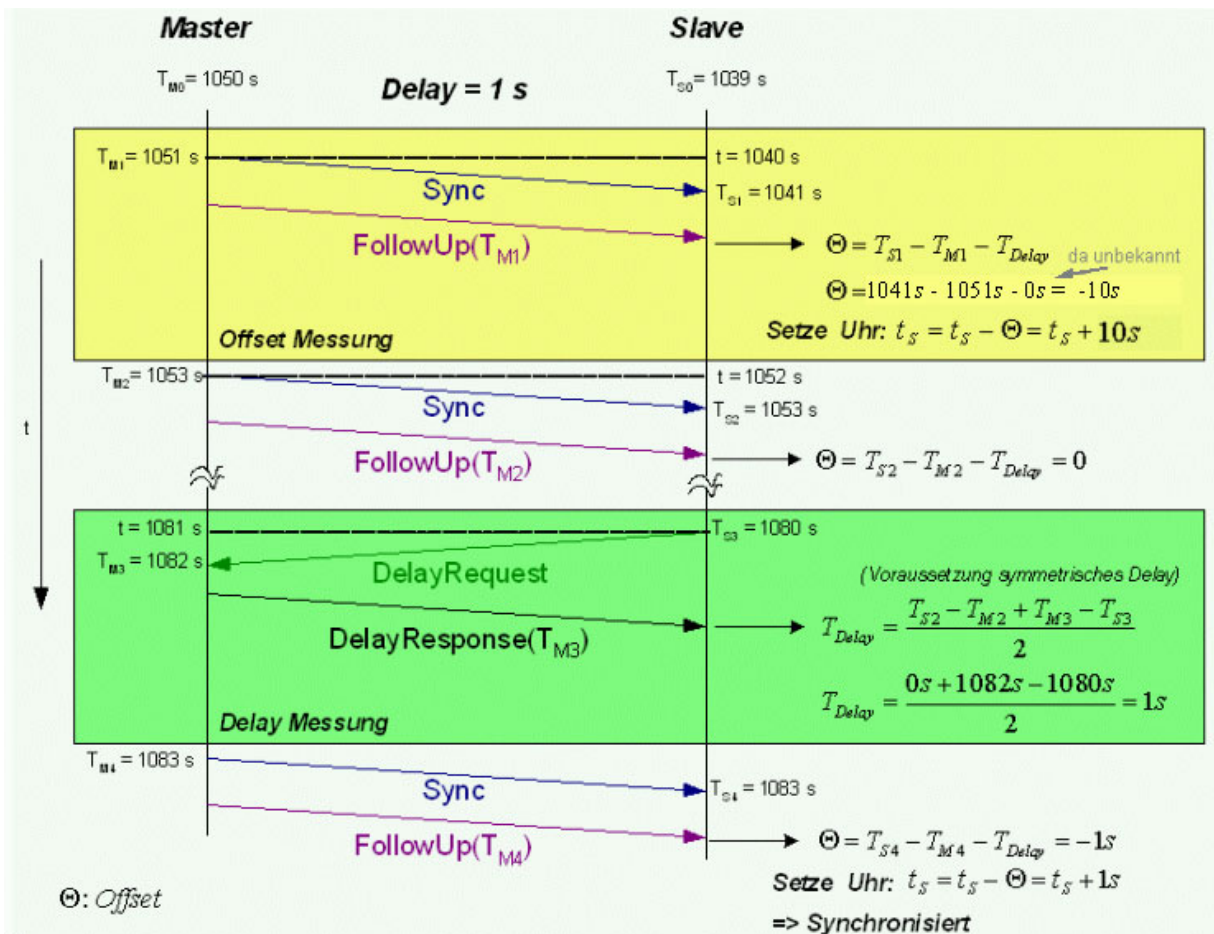


Abbildung 5: Synchronisationsalgorithmus

Automatische Masterwahl – Best Master Clock Algorithmus

Das Precision Time Protocol bestimmt Master- Slave Beziehungen zwischen kommunizierenden PTP Synchronisationselementen. Die Bestimmung der Zustände (Master/Slave) obliegt dem Best Master Clock (BMC) Algorithmus, welcher die Eigenschaften (Genauigkeit, Stratum, Drift, Varianz, ...) der kommunizierenden Uhren vergleicht und daraus die Zustände für alle lokalen Ports des PTP Elements ableitet. Jedes PTP Element führt lokal den BMC Algorithmus aus und bestimmt somit den Zustand aller lokalen Ports. Die aktuellen Eigenschaften der Master Uhr übermitteln diese in Synchronisationsnachrichten zyklisch an die Slaves. Der Vorteil dieses Algorithmus besteht darin, dass die Knoten diese Zustände nicht aushandeln müssen, sondern in jedem Knoten individuell berechnen. Dadurch wird gewährleistet, dass sich das PTP Netzwerk automatisch zu einer Baumstruktur konfiguriert, ausgehend von der Besten zur Verfügung stehenden Uhr, dem Grandmaster. Eine Konfiguration mit mehreren oder keinem Master sowie instabile Zustände werden durch diesen Algorithmus verhindert.

Anhang 2 – Die Implementierung von PTP im Detail

Das Besondere an der Architektur von PTP ist die Trennung des zeitkritischen Teils, der in Hardware implementiert ist und dem von harten Zeitbedingungen entkoppelten Protokoll, dem Softwareteil. Somit läuft das Protokoll in einem niedrigeren Prozess beziehungsweise auf einem Prozessor mit geringen Leistungsanforderungen.

Die Hardware Einheit besteht aus einer hochpräzisen Echtzeit-Uhr und einer Zeitstempereinheit (TSU) zum generieren der Zeitstempel. Die Software Komponente implementiert das eigentliche IEEE1588 Protokoll mit der Anbindung an die Echtzeit-Uhr und der HW Zeitstempereinheit. Abbildung 6 illustriert das Zusammenwirken der Hard- und Softwarekomponente eines IEEE1588 Synchronisationselements.

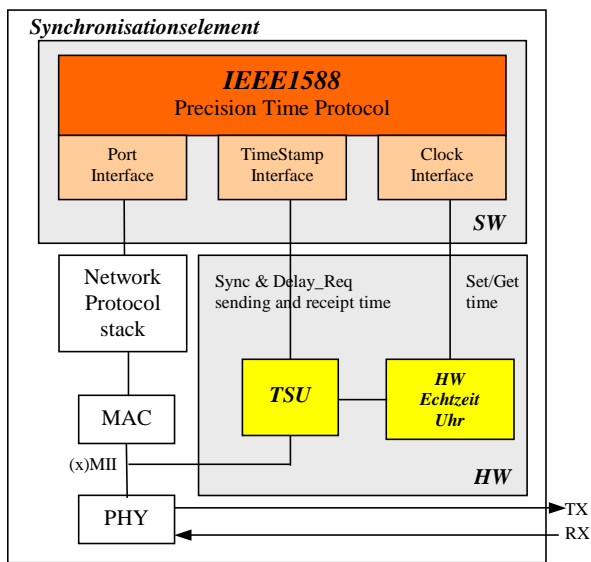


Abbildung 6: Komponentenbild eines Synchronisationselement

TSU - TimeStamp Unit

Wichtig für eine generische, portierbare Implementierung des PTP Protocols ist eine möglichst betriebssystemunabhängige Modellierung der Softwarekomponenten. Um dies zu erreichen, hat Hirschmann in seiner Lösung drei Schichten mit unterschiedlichem Abstraktionslevel eingeführt. Der **Protokoll Layer implementiert das** vom Betriebssystem unabhängige Precision Time Protocol. Der **OS Abstraction Layer** bildet die Schnittstelle zwischen PTP und dem verwendeten Betriebssystem. Die vom Betriebssystem bereitgestellten Funktionen – Tasks / Prozesse, Semaphoren, Timer, Sockets, etc. – werden über den **OS Layer** eingebunden. Die folgende Abbildung zeigt die Interaktion der einzelnen Schichten.

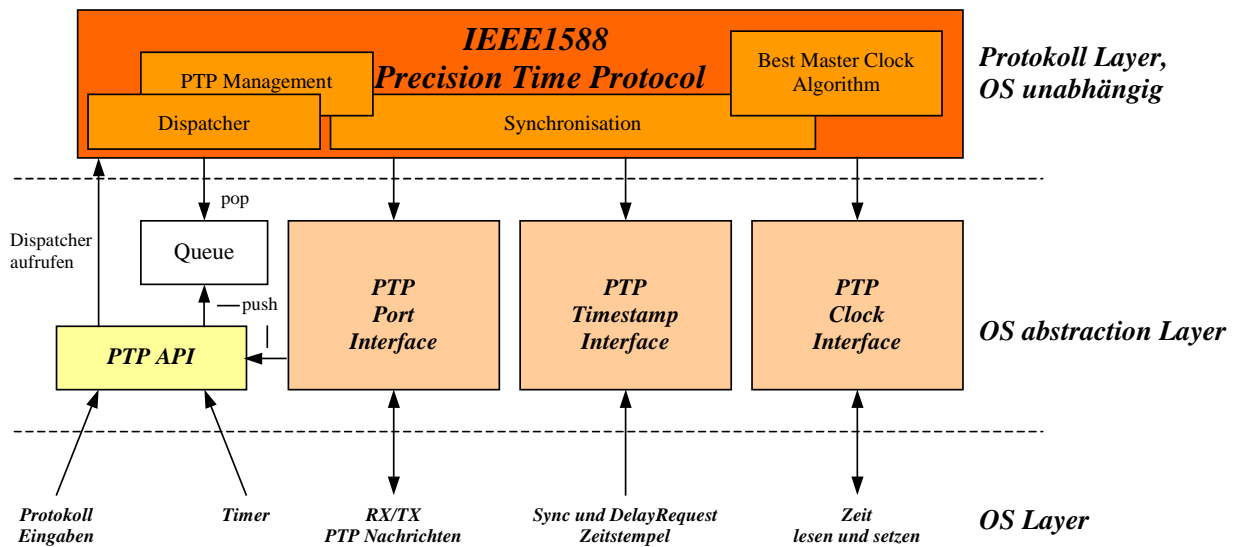


Abbildung 7: Interaktionsdiagramm

Die oberste Schicht implementiert PTP zur Synchronisation von Uhren in einem Netzwerk und kann auf unterschiedlichen Kommunikationselementen (PC, Switch, Router, etc.) eingesetzt werden. Hier befindet sich die eigentliche Intelligenz zum Synchronisieren der einzelnen Kommunikationselemente. Innerhalb des **Protokoll Layers** wurden nur ANSI/ISO C konforme Funktionen verwendet. Somit wird sichergestellt, daß man das Protokoll ohne tiefe Eingriffe in die Funktionsweise auf verschiedene Plattformen portieren kann. Ein Protokoll Dispatcher gewährleistet die atomare Ausführung von Funktionen in einem einzelnen Prozess. Die Kommunikation zwischen dem Protokoll und dem OS Abstraction Layer erfolgt über eine Queue und drei definierte Schnittstellen.

Die mittlere Schicht kapselt Betriebssystem abhängige Funktionen, die man bei Bedarf anpassen muß.

Das **Timestamp Interface** stellt dem Precision Time Protocol die erfaßten Zeitstempel für Sync und DelayRequest Nachrichten zur Verfügung. Je nach Ausbaustufe (Präzisionsanforderung) generiert die Zeitstempel entweder eine HW Einheit (TSU) oder aber die Software selber. „Software Zeitstempel“ werden am besten im Betriebssystem abhängigen NIC Treiber (RX-ISR, Sende Prozess) so nah wie möglich am Transportmedium generiert.

Über das **Clock Interface** liest und modifiziert man die lokale Uhr. Diese Funktionen muß man ebenfalls je nach Plattform anpassen. Realisierungen, welche keine Echtzeit-Uhr zu Verfügung stellen, verwenden die System Uhr des Betriebssystem oder optimierte Lösungen wie beispielsweise den Nanokernel unter UNIX Derivaten. Neben der Ansteuerung der lokalen Uhr beinhaltet diese Schnittstelle die Regelalgorithmen, welche für die Güte [Genauigkeit, Stabilität, Einschwingverhalten, etc.] der Zeitsynchronisation verantwortlich sind.

Das **Port Interface** wird dazu verwendet, PTP Nachrichten zu versenden beziehungsweise zu empfangen. Die IEEE1588 Telegramme benutzen ausschließlich UDP/IP Multicast Pakete und ermöglichen es somit, daß man sie über die Socket Schnittstelle des IP Protokoll-Stacks senden und empfangen kann. Zeitliche

Anforderungen darf man vernachlässigen, da die Zeitstempel direkt am Transport Medium generiert werden.

Die Eingaben an das Protokoll (Konfiguration, Diagnose, PTP Telegramme) laufen über die **PTP API**.

Geringer Portierungsaufwand, bei Umstieg auf eine andere Plattform

Durch diese Vorgehensweise wurde erreicht, daß man bei einer Portierung des Protokolls auf eine andere Plattform nur die untersten zwei Schichten anpassen muß, wodurch das eigentlich Protokoll (~ 90 - 95 % der Quelltexte) unberührt bleibt.

Der Softwareteil der beschriebenen PTP Referenz Architektur wurde in objektbasiertem C umgesetzt. Dadurch konnten trotz Verwendung der Programmiersprache C die Vorteile der objektorientierten Analyse/Design/Programmierung angewendet werden: Methodik, Kapseln, Wiederverwendbarkeit, etc.

Um die leichte Portierbarkeit der Software zu prüfen, hat Hirschmann Referenzimplementierungen von Ordinary und Boundary Clocks unter den Plattformen Linux, VxWorks und Windows durchgeführt.

Die Präzision einer Softwareimplementierung

Nach Fertigstellung der Arbeiten wurde zunächst untersucht, welche Präzision mit einer *reinen Software Implementierung* erreicht werden kann. Messungen sowohl auf Linux- Plattformen als auch auf Windows und VxWorks Plattformen ergaben eine typische *Synchronisationsgenauigkeit* von $\pm 10\mu\text{s}$ bis $\pm 200\mu\text{s}$.

Die erzielte Genauigkeit ist dabei im wesentlichen abhängig von der verwendeten Plattform, der Netzlast, Systemauslastung / interne Buslast und dem Ort, an dem der Zeitstempel generiert wird. Je hardware-naher die Zeitstempelung erfolgt, desto besser die Präzision.

Hardware für höchste Präzision

Für Synchronisationsgenauigkeiten im Submikrosekunden- Bereich genügt eine Softwareimplementierung nicht, deshalb müssen hier die erforderlichen Zeitstempel in einer sogenannten Hardware Zeitstempelinheit direkt am Transport Medium generiert werden.

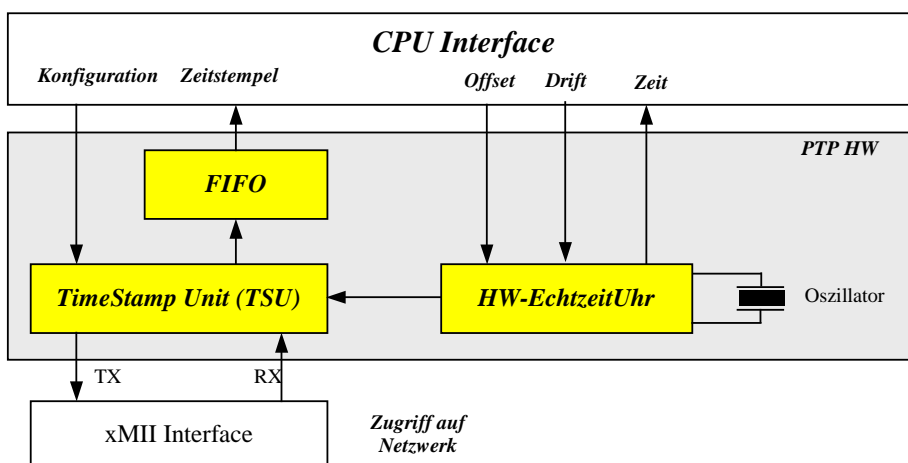


Abbildung 8: Hardware Blockschaltbild

Die **Timestamp Unit** (TSU) wurde ebenfalls in Hardware umgesetzt. Sie setzt dabei auf der Media Independent Interface (MII) - Schnittstelle auf. Erkennt die TSU eine *Sync* oder *DelayRequest* Nachricht, so vermerkt sie den Zeitpunkt des Auftretens. Des weiteren extrahiert sie Informationen zur eindeutigen Identifizierung der Nachricht. Diese Informationen puffert die TSU in einem FIFO und ermöglicht es somit, den erfaßten Zeitstempel der korrespondierenden PTP Nachricht später zuzuordnen.

Die absolute Zeit des Synchronisationselements verwaltet die **HW Echtzeit- Uhr** und dient als sehr präzises Zeitmetronom. Die HW Uhr zählt hierbei Oszillator Ticks, welche der Clock Treiber in Sekunden und Nanosekunden umrechnet. Integraler Bestandteil der digitalen Echtzeit-Uhr, ist die Offset (Phasen-) und Drift (Frequenz-) Korrekturereinheit. Sie ermöglicht die Justierung einer Echtzeit-Uhr im Submikrosekundenbereich auf eine Referenz-Uhr. Die Separierung der Phasen und Frequenz Einheit ermöglicht den gezielten Stelleingriff der Regelung, wodurch eine schnelle Konvergenz des Systems erzielt wird. Das System kann somit Phasensprünge – hervorgerufen z. B. durch Veränderung der Topologie – unabhängig von der Driftkorrekturereinheit ausregeln. Dies hat den Vorteil, daß die lokale Uhr die Frequenz nicht verändern muß und somit ein Phasensprung schnell kompensiert ist.

Die **Offset Kompensation** gleicht den absolut Wert der lokalen Uhr zur Referenz-Uhr ab.

Die **Driftkorrektur** bewirkt die Frequenz Synchronisation der lokalen Echtzeit-Uhr mit der Referenz-Uhr. Damit erreicht man, dass die lokale Uhr mit der Frequenz des Masters „tickt“. Die Herausforderung besteht nun darin die relative Drift δ der lokalen Uhr zur Referenz Uhr zu bestimmen. Folgende Gleichung kann vereinfacht für die Driftkorrektur angenommen werden.

$$f(t)_{Slave} = f(t)_{Master} + \delta + \varepsilon_{Quantisierung}$$

Wobei $f(t)$ die gegenwärtige Frequenz der entsprechenden Uhr und $\varepsilon_{Quantisierung}$ den Quantisierungsfehler darstellt.

Die gesamte Hardware läßt sich im ersten Ansatz in einem FPGA realisieren, wobei die Realisierung für einen Port etwa 20.000 bis 30.000 Logic Array Gates benötigt.

Die Randbedingungen für hoch präzise Zeitsynchronisation

Hinsichtlich der Synchronisationsgenauigkeit definiert PTP die Mindestanforderungen einer lokalen Uhr und gibt Empfehlungen zur Erhöhung der Präzision. Dabei sind folgende Aspekte zu beachten.

- Betriebssystem und Network **Stack Jitter**

Jitter innerhalb eines Synchronisationselements vermeiden Zeitstempel in Hardware.

- **Latenz** zwischen Master und Slave

Symmetrische Latenz (Hin- und Rückrichtung identisch) zwischen Master und Slaves kompensiert die Delay Messung. Eine Unsymmetrie bewirkt aber eine nicht kompensierbare Zeitdifferenz zwischen Master und Slave.

- **Jitter** in **Netzkoppelementen** (Repeater, Switch, Routers)

Jitter Untersuchung mit Netzkoppelementen haben gezeigt, dass der Jitter von Hubs bzw. Repeater zu vernachlässigen ist. Betrachtet man jedoch Switche im Hochlast- oder Überlastfall so besitzen diese einen nicht unerheblicher Verzögerungsjitter. So sind Werte bis zu einer Millisekunde keine Seltenheit. Der Jitter in einem Switch ist abhängig von der Nachrichtenlänge und der Anzahl der Pakete die schon gepuffert sind. Selbst wenn nur ein einziges langes Paket im Switch dem Synchronisationspaket zuvor gekommen ist, kann dies einen Jitter bis zu 125µs hervorrufen. Deshalb wird für ein hoch präzises Netz empfohlen, Switches mit der Boundary Clock Funktion auszurüsten. So stellt der Anwender sicher, daß die Synchronisation immer nur über eine Punkt zu Punkt Verbindung erfolgt, d. h. über eine physikalische Verbindung, deren Übertragungsjitter dann zu vernachlässigen ist.

- **Auflösung** und **Stabilität** der **lokalen Uhr**

Befolgt man die obigen Gesichtspunkte, so bestimmt die lokale Uhr des Synchronisationselements überwiegend die Synchronisationsgenauigkeit. Erster limitierender Faktor ist die Auflösung der lokalen Uhr, welche durch die Quarzfrequenz des Oszillators bestimmt wird. Der entscheidende Aspekt ist jedoch der Stabilität insbesondere der Kurzzeitstabilität des Oszillators zuzuschreiben. Die Güte der Synchronisationsgenauigkeit im eingeschwungen Zustand (Drift kompensiert), ist somit abhängig von der relativen Drift zwischen Master und Slave zwischen zwei aufeinanderfolgenden Synchronisationstelegrammen.

Regelung, das Unterscheidungsmerkmal

PTP beschreibt das Verfahren mit dem Master und Slave ihre zeitliche Differenz (Offset) bestimmen. Das Protokoll kann diese Größe aber nicht direkt setzen. Verschiedene Algorithmen müssen herangezogen werden um den Offset und die Drift zu bestimmen beziehungsweise den Wert zu verifizieren. Je höher die Präzision der lokalen Uhr zur Referenz Uhr, desto mehr muß man diese Algorithmen ausfeilen. Vereinfacht könnte man die Verifikation des Offsets z.B. dadurch erreichen, indem man den neuen Wert mit den vorhergehenden vergleicht. Wenn dieser dann einen bestimmte Differenz überschreitet, geht man davon aus, daß er ungültig ist. Man muß dabei aber berücksichtigen, daß sich tatsächlich etwas geändert haben könnte und sollte die neuen Werte dann doch akzeptieren, wenn sie mehrfach hintereinander kommen.

Die Driftkompensation (Frequenz Synchronisation) ist etwas komplexer und wird auch über einen größeren Zeitraum durchgeführt.

Die Güte der Implementierung der Regelung wird zukünftig die Anbieter differenzieren. Die Anwender können zur Bewertung einer Regelung folgende Kriterien heranziehen.

- Das Einschwingverhalten eine Regelung charakterisiert deren Güte. Die ausschlaggebende Größe hierfür ist die Einschwingzeit. Das ist die Zeitdauer, welche eine lokale Uhr benötigt, um sich auf die Referenz Uhr (Frequenz und Phase) zu synchronisieren.
- Störnunempfindlichkeit und Stabilität einer eingeschwungenen Uhr. Ist die Uhr synchronisiert, so sollte die Regelung eine gewisse Trägheit besitzen und nicht auf minimale Ausreißer mit hohem Stelleingriff des Reglers entgegen zu wirken. Dennoch muss die Regelung auf dauerhafte Änderungen

(Topologieänderung oder Masterwechsel) zügig reagieren, um die Synchronisation wieder herzustellen. Dies wird beispielsweise durch eine adaptive Regelung erreicht.

Ergebnisse

In einem Laboraufbau wurden im ersten Ansatz zwei Ordinary Clocks direkt miteinander verbunden, die in Hirschmann Ethernet Switches der MICE Familie implementiert wurden. Um das Protokoll auf Herz und Nieren zu testen wurde zusätzlich mit einem Generator eine hohe Netzlast angelegt.

Damit die Differenz zwischen lokaler Uhr und Referenzuhr möglichst anwendungsnah überprüfen werden konnte, wurden beide Einheiten mit einem Pulse Per Second (PPS) Signalausgang versehen und auf ein Oszilloskop geführt (Abbildung 9). Damit konnten nun sehr elegant die Abweichung der beiden Signale erfasst und auch die Häufigkeitsverteilung der Abweichung dargestellt werden.

Die Synchronisationsgenauigkeit die erreicht wurde lag dabei bei ± 60 ns (max. Jitter). Die Messung lief dabei über einen Zeitraum von 84 Stunden.

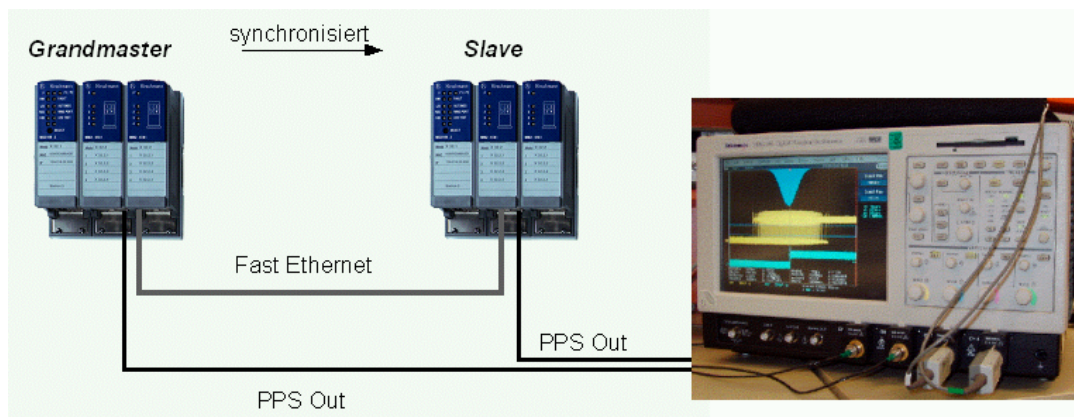


Abbildung 9: Der Meßaufbau

Abbildung 10 zeigt die Häufigkeitsverteilung der Offset Werte zwischen Master und Slave in Nanosekunden. Die Standardabweichung beträgt 14,9 ns und der Mittelwert 4,2 ns.

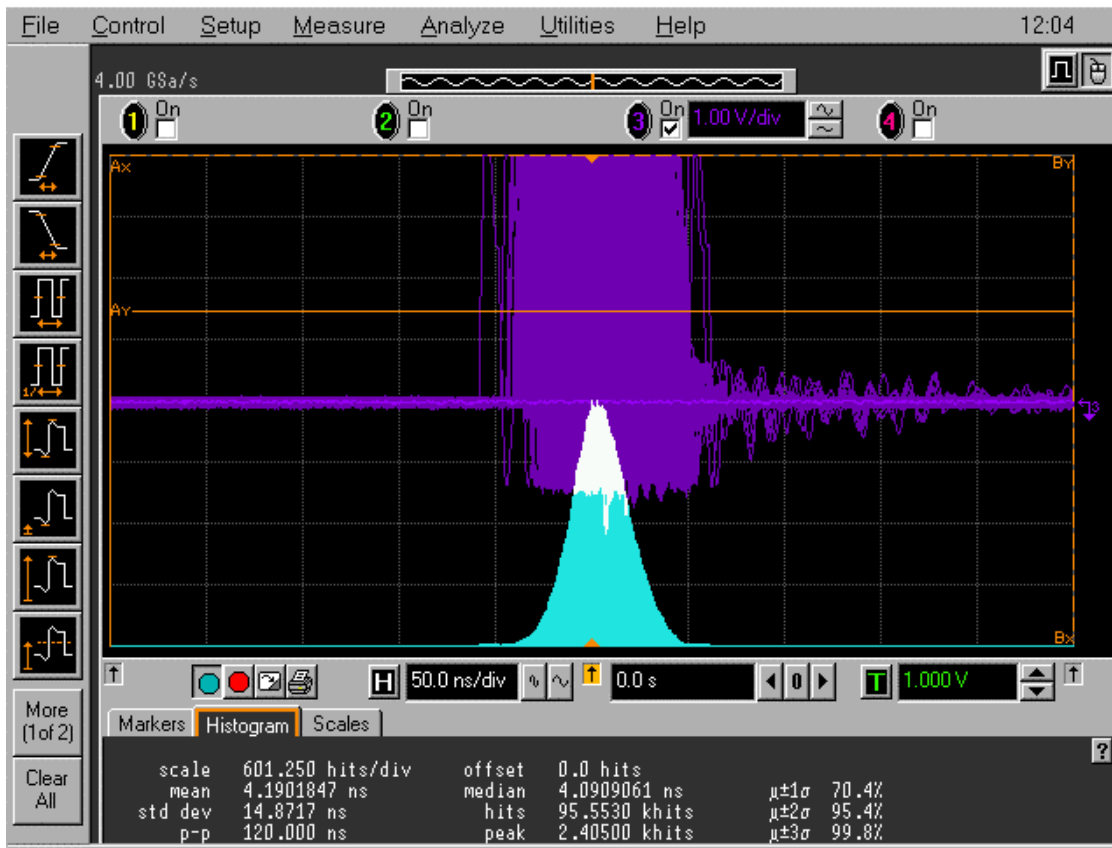


Abbildung 10: Erreichte Genauigkeit

Die Drifteinstellung der Oszillatoren beschränkt bei den vorliegenden Prototypen die Synchronisationsgenauigkeit. Eine Quarzfrequenz von 50 MHz (± 50 ppm) ergibt eine Auflösung von 20 ns. Somit kann das System die Drift nur im Bereich von ± 20 ns pro Sekunde einstellen. Betrachtet man nun noch die relative Drift zwischen der lokalen Uhr und der Master Uhr während zwei aufeinanderfolgenden Synchronisationstelegrammen, so wird schnell deutlich, dass die Kurzzeitstabilität der Oszillatoren die Synchronisationsgenauigkeit im eingeschwungenen Zustand wesentlich bestimmt.

Andreas Dreher, Dirk Mohl
 Hirschmann Automation and Control GmbH
 Neckartenzlingen, Germany