

Precision Time Protocol (PTP)

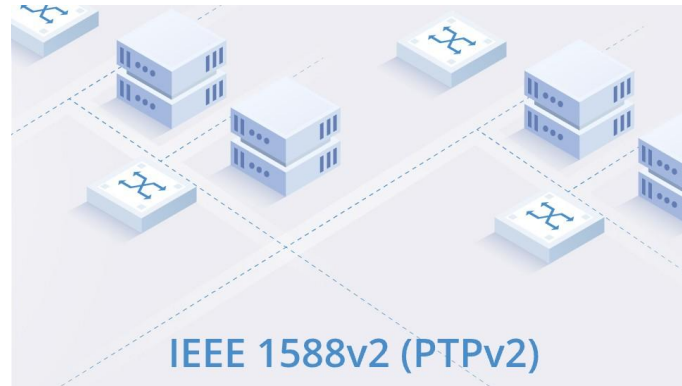
Gruppe 7:
Malte Kuehn, Chin-I Feng

HTWG **Agenda**

- Einleitung
- PTP im OSI-Modell
- Konzepte von PTP
- Ablauf von PTP
- Beispiel
- Anwendungen
- Pro und Contra

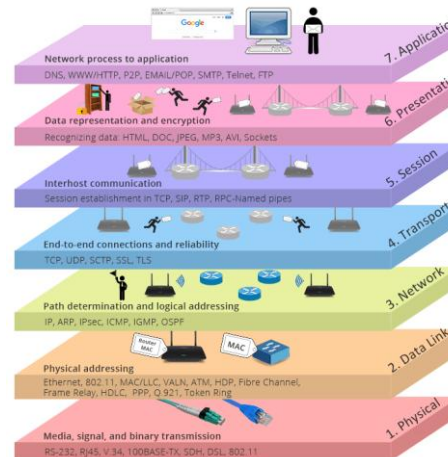
HTWG Einleitung

- **Precision Time Protocol Version 2 (PTPv2)** oder **IEEE 1588v2** ist ein **Netzwerkprotokoll** zur hochgenauen Zeit-Synchronisation in Computernetzwerken.
- Fokus auf lokalisierte Netzwerke und höhere Genauigkeit.
- Genauigkeit in lokalen Netzwerken:
 - Nanosekundenbereich in der Hardware-Version
 - Mikrosekundenbereich in der Software-Version



H T W E PTP im OSI-Modell G I

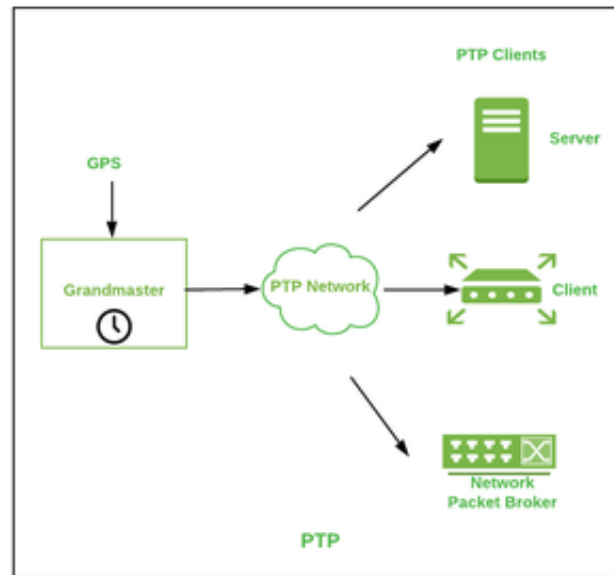
- **Primärschicht:** Anwendungsschicht (Layer 7)
 - Dient der Synchronisation von Uhren in Netzwerkgeräten
- **Verwendete Transportschicht:** UDP (Layer 4) für die Nachrichtenübertragung
 - Port 319: Für Event-Nachrichten (z.B. *Sync*, *Delay_Req*).
 - Port 320: Für allgemeine Nachrichten (z.B. *Follow_Up*, *Delay_Resp*).
- **Netzzugang:** Enge Zusammenarbeit mit Layer 2 (Ethernet) bei hardwarebasierter Implementierung



Konzepte von PTP

Master-Slave-Architektur

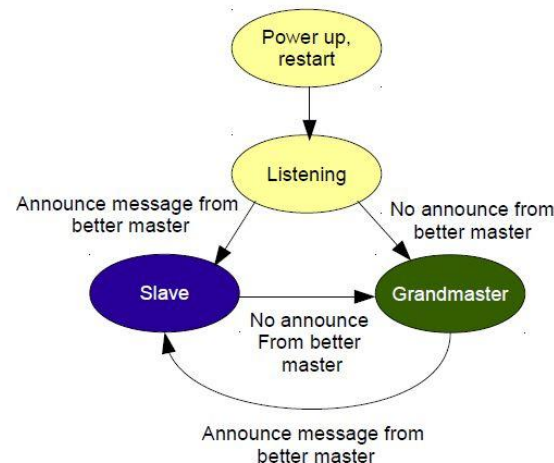
- **Grandmaster Clock** agiert als primäre Zeitquelle und gibt die genaue Uhrzeit vor.
- **Slave-Clocks (PTP-Clients)** synchronisieren sich mit dem Grandmaster Clock, um eine einheitliche Zeit zu gewährleisten.



Konzepte von PTP

Netzwerkorganisation

- Ein PTP-Netzwerk besteht aus kommunizierenden Uhren
- **Best Master Clock Algorithmus (BMCA) :**
 - Ermittelt die Uhr (Grandmaster Clock), die die präziseste Zeit liefert
 - Bewertet Uhren basierend auf mehreren Eigenschaften (z.B. Qualität, Priorität, Varianz)
- Erhöhung der Genauigkeit eines PTP-Netzwerks, indem er **automatisch** die beste Zeitquelle auswählt und bei Änderungen im Netzwerk **dynamisch** reagiert.

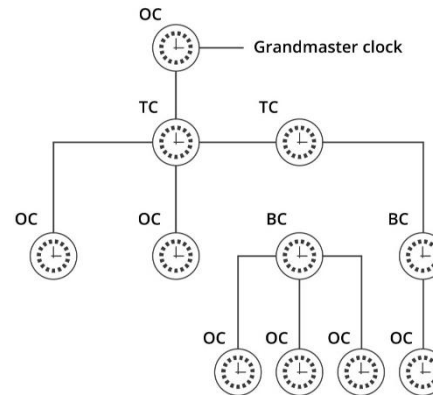


H T W E G I

Konzepte von PTP

Uhrtypen

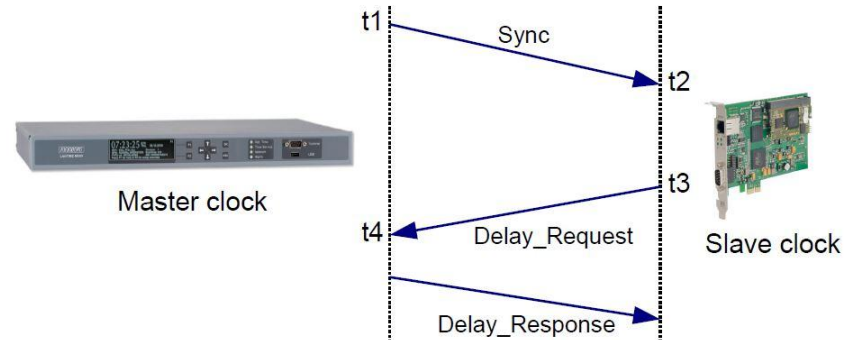
- **Ordinary Clock (OC):**
 - Kann entweder als Quelle (Master) oder Empfänger der Zeit (Slave) fungieren
- **Boundary Clock (BC):**
 - Erfüllt sowohl Master- als auch Slave-Rollen
 - Transportiert Zeitinformationen über Netzwerkgrenzen hinweg (z. B. Router)
- **Transparent Clock (TC):**
 - Verbessert die Übertragung von Zeitinformationen innerhalb eines Netzwerks
 - Empfängt und korrigiert PTP-Nachrichten zur Minimierung von Verzögerungen



Konzepte von PTP

1-Step-Modus

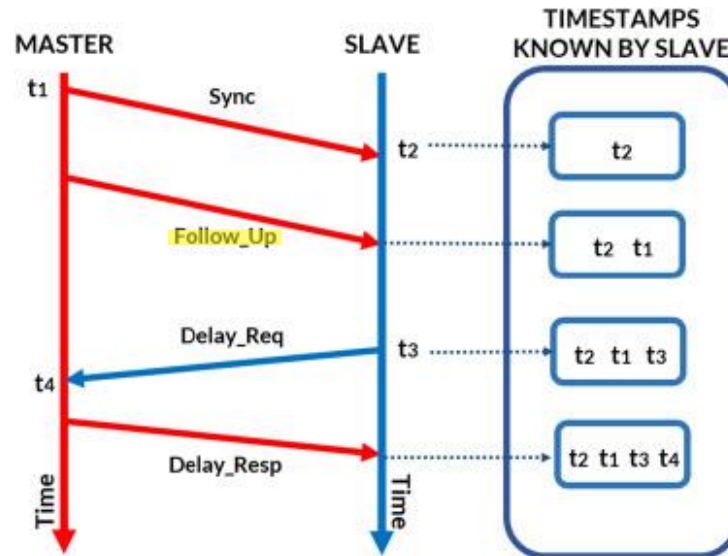
- Der Master sendet eine *Sync*-Nachricht mit einer **exakten Sendezeit** an die Slave-Clocks
- Erfordert spezielle Hardware, die die exakte Zeit direkt beim Senden hinzufügen kann
- Weniger Nachrichten und Schnellere Synchronisation, aber **hardwareabhängig**



Konzepte von PTP

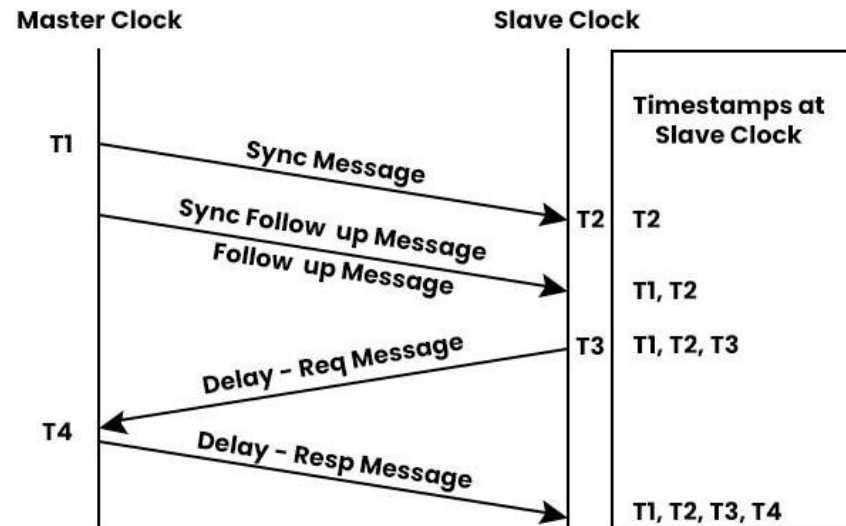
2-Step-Modus

- Die exakte Sendezeit wird **nicht** in der *Sync*-Nachricht selbst gesendet
- Stattdessen wird die genaue Sendezeit in einer separaten *Follow_Up*-Nachricht nachgeliefert
- Flexible Implementierung, jedoch mit leicht **höherer Latenz** im Vergleich zum 1-Step-Modus



Ablauf von PTP Sync-Nachricht

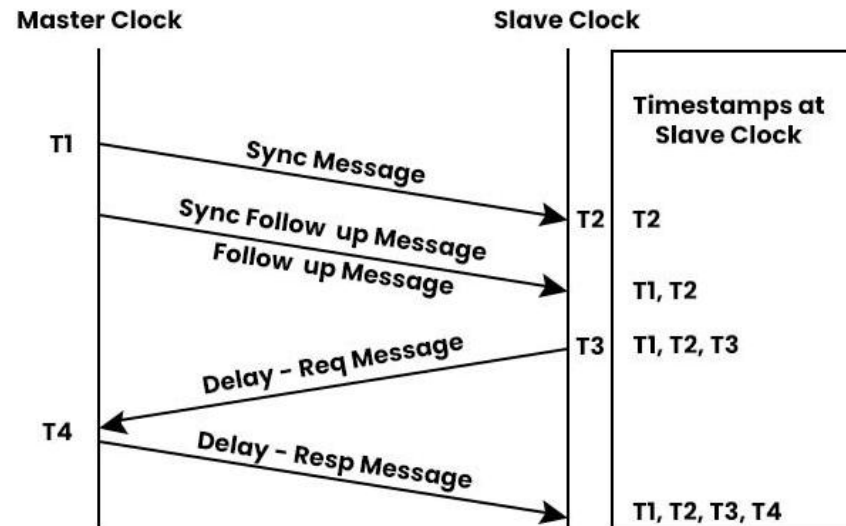
- Der Master Clock sendet eine *Sync*-Nachricht mit einem groben Zeitstempel an die Slave Clocks.
- Dieser Zeitstempel entspricht **nicht** der exakt gemessenen Sendezeit.
- Diese Nachricht dient dazu, den Synchronisationsprozess einzuleiten und die grobe Zeitinformation schnell an die Slave Clocks zu übermitteln.



Ablauf von PTP

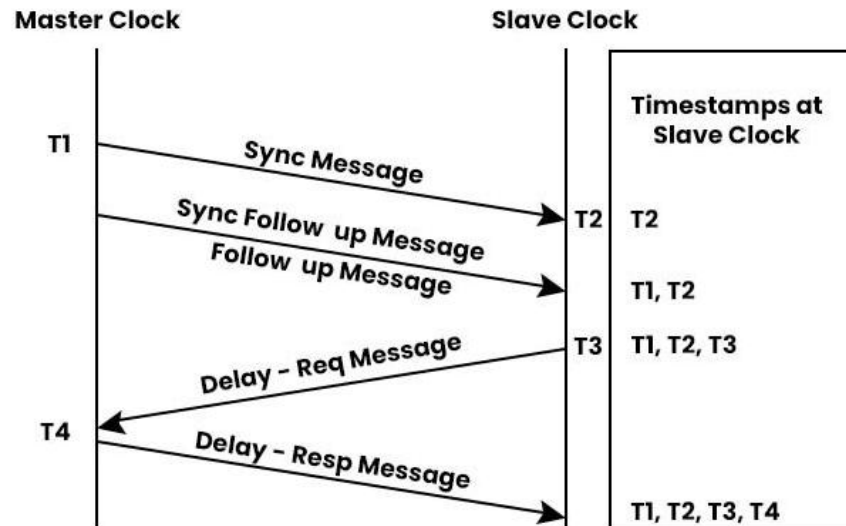
Follow_Up-Nachricht

- Die *Follow_Up*-Nachricht wird vom Master Clock gesendet, um die **exakte Sendezeit** der zuvor gesendeten *Sync*-Nachricht zu übermitteln.
- Diese Nachricht ergänzt den groben Zeitstempel der *Sync*-Nachricht und ermöglicht dadurch eine **präzisere Synchronisation** der Slave Clocks.



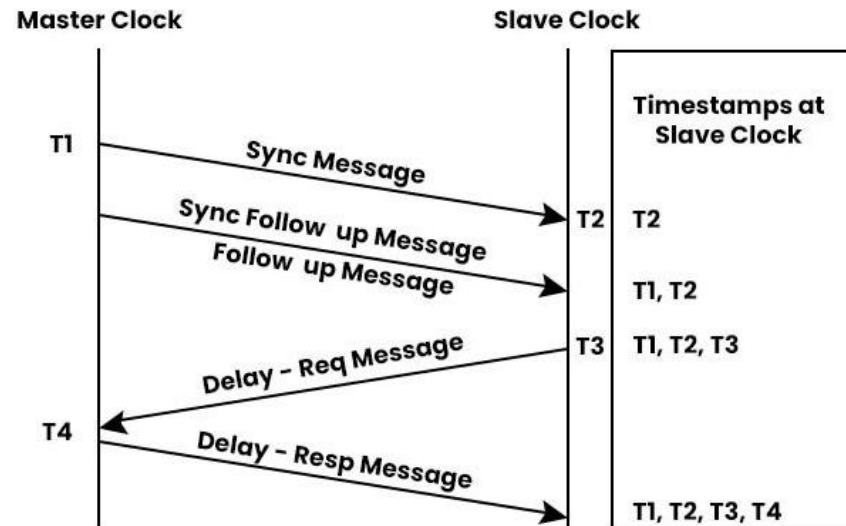
Ablauf von PTP Delay_Request-Nachricht

- Die *Delay_Request*-Nachricht wird von der Slave Clock an die Master Clock gesendet, um die Verzögerungszeit im Netzwerk zu messen.
- Der Slave versieht diese Nachricht mit einem Zeitstempel, der den Zeitpunkt ihres Versands angibt.
- Die Master Clock nutzt diesen Zeitstempel, um die **Netzwerklatenz** zu berechnen.

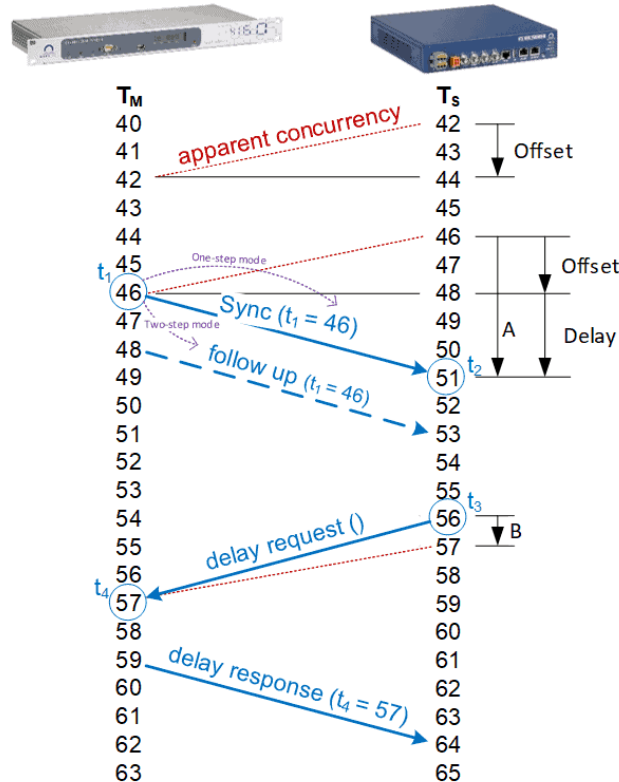


Ablauf von PTP Delay_Response-Nachricht

- Die *Delay_Response*-Nachricht wird von der Master Clock als Antwort auf die *Delay_Request*-Nachricht gesendet.
- Sie enthält den exakten Zeitstempel des Empfangs der *Delay_Request*-Nachricht, den die Slave Clock nutzt, um die Netzwerklatenz zu berechnen.



Beispiel



Offset and delay calculations

Theory:

$$A = t_2 - t_1 = \text{Delay} + \text{Offset}$$

$$B = t_4 - t_3 = \text{Delay} - \text{Offset}$$

$$\text{Delay} = \frac{A + B}{2}$$

$$\text{Offset} = \frac{A - B}{2}$$

Example:

$$A = 51 - 46 = 5$$

$$B = 57 - 56 = 1$$

$$\text{Delay} = \frac{5 + 1}{2} = 3$$

$$\text{Offset} = \frac{5 - 1}{2} = 2$$

- **Telekommunikation:** Notwendig für die Synchronisation von Mobilfunknetzen.
- **Finanztransaktionen:** Vermeidung von Zeitstempelfehlern bei Börsengeschäften
- **Energiewirtschaft:** Verhindert Zeitabweichungen in Smart Grids.
- **Industrielle Automatisierung:** Synchronisierung von Robotik und Fertigungslinien



• Pro

- **Hochpräzise Zeit-Synchronisation:** Bietet eine Genauigkeit im Nanosekundenbereich
- **Flexibilität:** Unterstützt sowohl software- als auch hardwarebasierte Implementierungen
- **Skalierbarkeit:** Funktioniert in Netzwerken mit einer großen Anzahl von Geräten
- **Integration:** PTP funktioniert über Ethernet, was die Integration in bestehende IT-Infrastrukturen erleichtert

• Contra

- **Kosten:** Implementierung von PTP mit hardwarebasierten Lösungen kann teuer sein
- **Komplexität:** Erfordert eine genaue Konfiguration und ein Verständnis der Netzwerkarchitektur

- <https://community.fs.com/blog/ieee-1588v2.html>
- <https://www.fs.com/de/blog/tcpip-vs-osi-whats-the-difference-between-the-two-models-5860.html>
- <https://www.geeksforgeeks.org/precision-time-protocol-ptp/>
- <https://blog.meinbergglobal.com/2013/11/14/makes-master-best/>
- <https://community.fs.com/blog/ieee-1588v2.html>
- <https://blog.meinbergglobal.com/2013/10/28/one-step-two-step/>
- <https://www.sciencedirect.com/topics/engineering/precision-time-protocol>
- <https://www.mobatime.com/article/ptp-precision-time-protocol/>
- <https://www.geeksforgeeks.org/clock-synchronization-in-distributed-system/>

**Vielen Dank
für die Aufmerksamkeit**