PTP* Introduction

- Part of workshop** on "HSR/PRP and PTP: Network Redundancy and Time Clock Synchronization" -

기안도
adki@future-ds.com

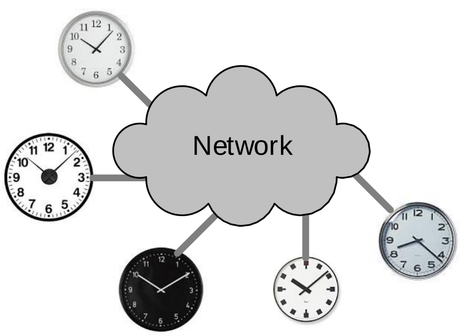
주최/주관: 한국통신학회 군통신연구회 / 명지대학교
장소: 숭실대학교 조만식기념관 427호
일자: 2019년6월7일

* PTP: Precision Time Protocol
** 이중화네트워크와 시각동기화 워크샵

Table of Contents

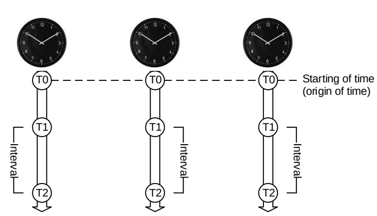
- Background
  - Clocks on the network
  - Distributed ideal clocks
  - Distributed clocks with variable delay network
  - Clock synchronization algorithms
  - PTP in short
  - PTP examples
  - PTP mathematics
  - Clock
  - Terminologies
  - Adjustment
  - Time transfer technologies
  - PTP related standards

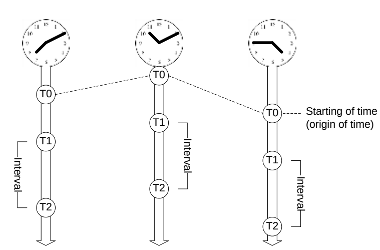
Clocks on the network



- **Objective**
  - Synchronize wall clocks, which are connected over packet-based network.
    - ➲ 패킷망에 연결된 시계들의 시간을 맞춘다
- **Implications**
  - '**Wall clock**' means real-time in nano/micro/milli second accuracy.
    - ➲ 실시간 정확도
  - '**Wall clocks over network**' mean a distributed system.
    - ➲ 분산시스템
  - '**Packet-based network**' means network delay varies.
    - ➲ 메시지(정보)를 주고 받음
    - ➲ 가변적 네트웍 지연

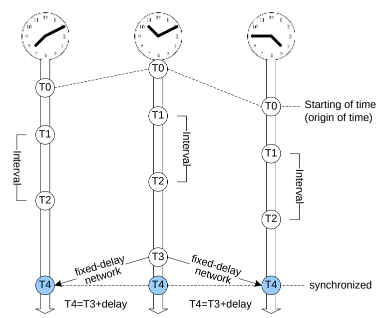
Distributed ideal clocks (1/2)





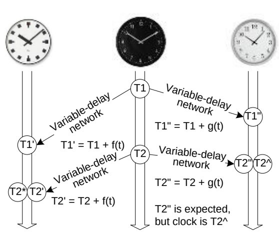
- Identical clocks with the same starting time.
  - There is nothing to do with synchronization.
    - ➲ 동일한 시작 시점을 갖는 '완전히 동일한 시계'들
      - 동기화가 필요치 않음
- Identical clock with different starting time.
  - There should be a mechanism to adjust offset.
    - ➲ One time synchronization is sufficient.
    - ➲ 시작 시점이 다른 '완전히 동일한 시계'들
      - 한 번만 시간을 맞추면 됨
        - ❖ 정확한 통신 지연 파악이 필요

Distributed ideal clocks (2/2)



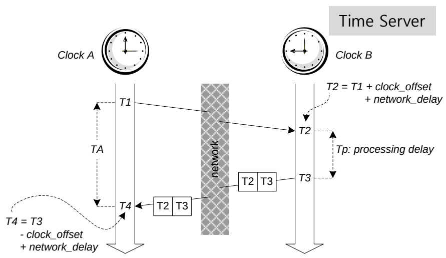
- Distributed clock can be synchronized by communication.
  - For identical clocks with different starting time
  - Need something about communication.
    - What if communication delay is not deterministic? → Need mechanism to measure communication delay.
    - 시작 시점이 다른 '완전히 동일한 시계'들
      - 한 번만 시간을 맞추면 됨
        - 정확한 통신 지연 파악이 필요
- What if clocks are not the same?
  - Each clock has different frequency and starting point
  - 만약 각 시계가 동일하지 않다면?
    - 각 시계가 주파수와 시작 시점이 다르다면?

Distributed clocks with variable delay network



- Actual (i.e., non-ideal) clocks
  - Phases differ
  - Frequencies differ
  - 실제 시계들은 위상과 주파수가 다름
    - 위상: 24시간 중 몇 시; 서기 몇년 몇월
    - 주파수: 1초에 해당하는 시간
- Variable-delay network based clock synchronization
  - Due to temperature, acceleration, aging and so on
  - 온도, 가속, 노화 등으로 네트웍 지연이 변함
- In addition to frequency calibration/correction of clocks, network-delay measurement is required.
  - 시계의 주파수 교정과 보정 뿐만 아니라 네트워크 지연도 측정해야 됨
- Delay and frequency should be measured periodically, since they are time-varying.
  - 네트워크 지연과 주파수는 주기적으로 측정하고 교정/보정 해야 됨

Clock synchronization algorithms (1/2)



- Cristian Algorithm
  - synchronization with a time server

Tdelay = (T4 – T1 – Tp) / 2
T2 = T1 + Toffset + Tdelay
T4 = T3 – Toffset + Tdelay
Toffset = [(T2 – T1) + (T3 – T4)] / 2
Tdelay = [(T4 – T1) – (T3 – T2)] / 2

F. Cristian, Probabilistic clock synchronization, Distributed Computing, Vol. 3, Issue 3, pp.146-158, Springer, 1989.

Clock synchronization algorithms (2/2)

- Berkeley Algorithm
  - synchronization without a time server

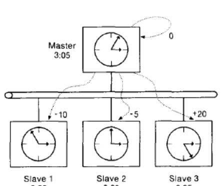


Fig. 1. The measurements.

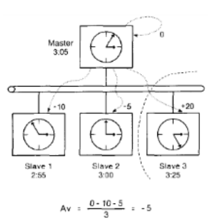


Fig. 2. The computation of the average.

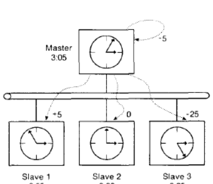


Fig. 3. The correction of the clocks.

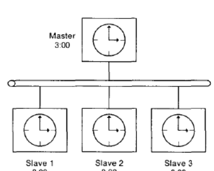
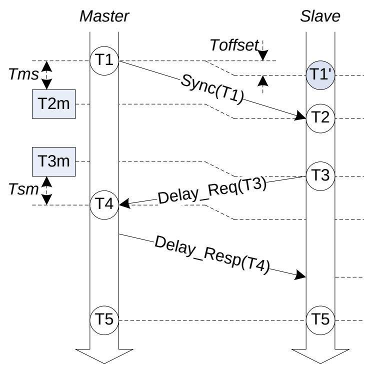


Fig. 4. Clocks are now synchronized.

R. Gusella and S. Zatti, The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3.BSD, IEEE Trans. on Software Engineering, Vol. 15, No. 7, pp.847-853, July 1989.

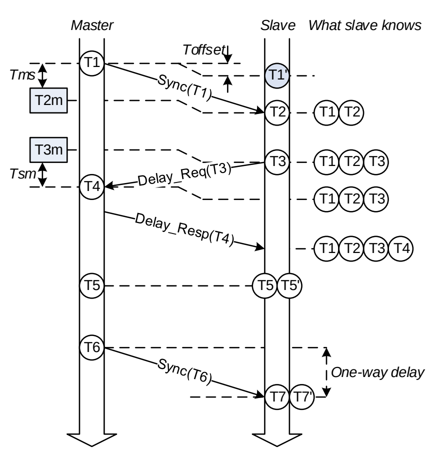
PTP in short (1/2)



Slave clock runs Toffset ahead

- 마스터와 슬레이브가 동일한 시계를 갖는다면 단순하게 통신지연만을 이용하여 동기 시킬 수 있다.
- 그러나 실제로는 시계가 차이가 나고, 이때 슬레이브 시계가 얼마나 빠른가를 Toffset이라 하며,
- Sync와 Delay_Req/Resp 패킷을 주고 받아서, 얻는 정보를 이용하여 통신지연과 Toffset을 유추해 낸다.
- 이때 통신지연은 양방향이 동일하다고 가정한다.
- 슬레이브 입장에서 T1' (즉, T1 + Toffset) 시점에 Sync가 네트웍에 전송되고, 슬레이브 시간 T2에 도착한다.
- 슬레이브는 역방향 통신지연을 측정하기 위해, T3에 Delay_Req를 전송하고, 마스터 입장에서 T4에 받아
- T4 정보를 담아서 Delay_Resp를 보낸다.
- 이렇게 되면 T1, T2, T3, T4 정보를 슬레이브가 갖게 되므로, 이들을 이용하여 Toffset과 Tdelay를 계산한다.

PTP in short (2/2)



[offset] Toffset = Tslave - Tmaster	Eq.1
[ms_difference] ΔTms = T2 - T1	Eq.2
[sm_difference] ΔTsm = T4 - T3	Eq.3
[ms_delay] Tms = T2m - T1	Eq.4
[sm_delay] Tsm = T4 - T3m	Eq.5

T1' is T1+Toffset;
T2 is T2m+Toffset;
T3 is T3m+Toffset;

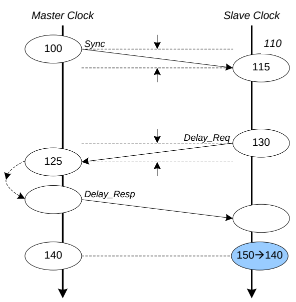
[ms_delay] Tms = T2m - T1 = T2 - Toffset - T1
= ΔTms - Toffset
[sm_delay] Tsm = T4 - T3m = T4 - (T3 - Toffset)
= ΔTsm + Toffset

Assume that Tms == Tsm,

One-way delay Tdelay = [(T2 - T1) + (T4 - T3)] / 2

Toffset = [(T2 - T1) - (T4 - T3)] / 2 = (T2 - T1) - Tdelay

PTP examples



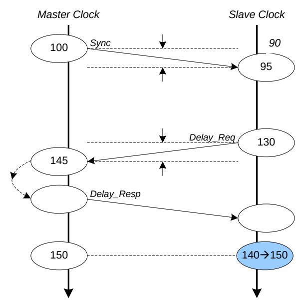
Delay = [(T2-T1)+(T4-T3)]/2
= [(115-100)+(125-130)] / 2
= [15 - 5] / 2 = 5

Offset = [(T2-T1)-(T4-T3)]/2
= [15 + 5] / 2 = 10

Slave has positive offset

Offset = 10

Delay = 5



Delay = [(T2-T1)+(T4-T3)]/2
= [(95-100)+(145-130)] / 2
= [-5 + 15] / 2 = 5

Offset = [(T2-T1)-(T4-T3)]/2
= [-5 - 15] / 2 = -10

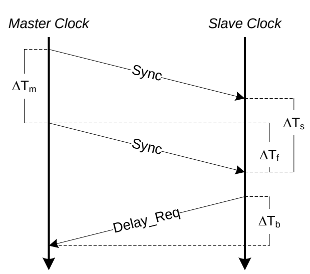
Slave has negative offset

Offset = -10

Delay = 5

Offset: Slave clock runs ahead of the master.

PTP mathematics



ΔT_m *Master interval*
ΔT_s *Slave interval*
ΔT_f *Forward delay*
ΔT_b *Backward delay*

$$\text{Delay } (T_d) = \frac{\Delta T_f + \Delta T_b}{2}$$

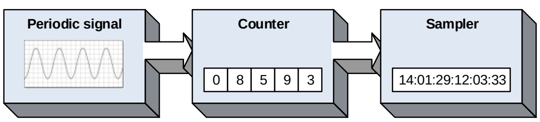
$$\text{Offset } (T_o) = \frac{\Delta T_f - \Delta T_b}{2} = \Delta T_f - T_d$$

$$\text{Drift } (\lambda) = \frac{\Delta T_m - \Delta T_s}{\Delta T_s}$$

- Offset: clock shift
  - 두 시계의 시점이 얼마나 다른가 (12시 ← →12시1분)
- Drift: clock period difference
  - 시계가 빠르고 느린 정도
- Slave clock is ahead of the master clock by the Offset (To).
  - decrease To
- Slave clock is slower than the master clock by the Drift (λ) per second
  - 슬레이브 클럭이 마스트 클럭보다 얼마만큼 느린가 (초당 몇 초 느린가)

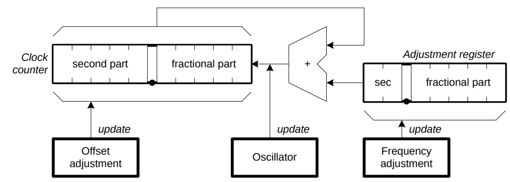
Clock

- Clock is a counter of periodic events.



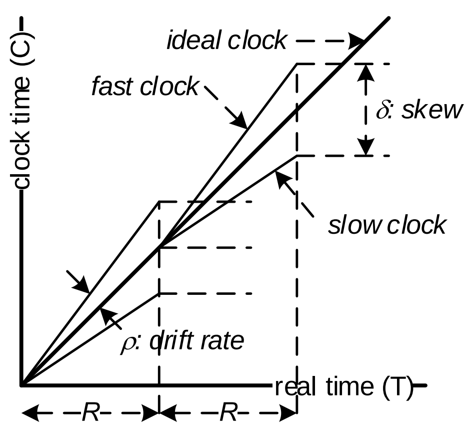
- VCO-based (Voltage Controlled Oscillator)
  - Rare implementation

- Counter-based clock



- Offset/phase can be adjusted by updating value of counter
  - 오프셋/위상은 카운트의 값을 변화시켜서 조정 가능
- Drifting/frequency can be adjusted by varying the amount of increment.
  - 드리프팅/주파수는 증가하는 값을 변화시켜서 조정가능

Terminologies



R: Resynchronization interval

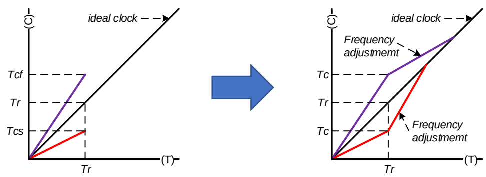
- Terminologies
  - Resynchronization interval R
  - Drift rate ρ
    - Clock drift: Count at different rates. (Different frequency of the oscillator.)

$$1-\rho \le \frac{dC}{dt} \le 1+\rho$$

    - Ideal clock: drifting rate 0
  - Clock skew δ
    - Clock skew (offset): Difference between time on two clocks.

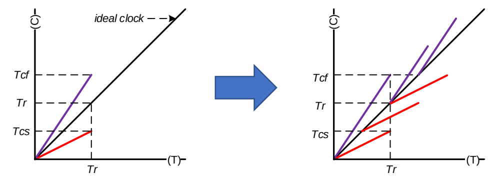
Adjustment

- Frequency adjustment



  - Syntonization
    - Frequency synchronization
      - different time, but keeping a constant time difference

- Offset adjustment



  - Synchronization
    - Phase synchronization
      - keep the same time

Time transfer technologies

	IRIG-B	(S)NTP	PTP	GPS
Accuracy (typical)	1-10μs	1ms – 10 ms	100ns-1μs	10ns
Transport media	Dedicated cables	Ethernet cable	Ethernet cables	Wireless
Protocol style	Master-slave	Client-server	Master-slave	
Built in latency correction	No	Yes	Yes	
Set-up	Configured	Configured	Self-organizing, or configured	
Update intervals	1 second	Minutes	1 second	
Specialized hardware	Required	No	Required	Required

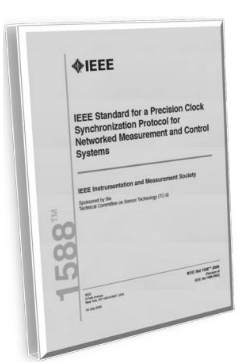
IRIG: Inter-Range Instrumentation Group; (S)NTP: Simple Network Time Protocol; GPS: Global Positioning System

PTP related standards

- IEEE 1588-2008
  - IEC 61588 Ed.2
    - IEC - International Electrotechnical Commission
  - Sub-microsecond synchronization
  - Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems
- IEEE 802.1AS
  - Standard for transport of precise timing and synchronization in Audio/Video Bridging (AVB) networks (AVB) networks.
  - It is based on IEEE 1588 V2, and includes a PTP profile.
  - It is also known as General Precision Time Protocol (gPTP).
- White Rabbit project
  - Sub-nanosecond synchronization
  - Utilizes Synchronous Ethernet (SyncE) to achieve syntonization and IEEE 1588 (1588) Precision Time Protocol (PTP) to communicate time
- (Not clear year)
  - IEEE 1588-2018 (?), PTPv2.1, PTPv3
    - Compatible with PTPv2
    - Security extensions
    - Sub-nanosecond accuracy

Table of Contents

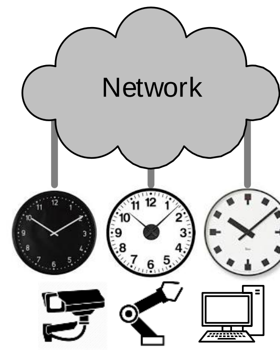
- Background
- IEEE 1588 PTP
- IEEE 1588 PTPv2 messages



- IEEE 1588 PTP
  - IEEE 1588 PTP
  - PTP v1 & V2
  - PTP implementation
  - PTP messages
  - Mechanisms to measure offset, delay, and drift
  - PTP clocks
  - PTP best master clock algorithm
  - PTP profiles

IEEE 1588 PTP

- IEEE 1588
  - (commonly known as Precision Time Protocol, PTP)
  - v1: 2002
  - v2: 2008 (IEC 61588 Ed.2)
- IEEE 1588 specifies "*A protocol to synchronize independent clocks running on separate nodes of a distributed measurement or control system to a high accuracy and precision*".
- IEEE 1588 is a protocol designed to synchronize real-time clocks in the nodes of a distributed system that communicate using a network.
  - IEEE 1588은 네트웍을 통해 통신하는 분산 시스템의 각 노드에 있는 시계를 동기시키는 프로토콜



Precision Time Protocol (PTP) is a protocol to synchronize clocks throughout a network and it achieves clock accuracy in the sub-microsecond range on a local area network.

PTP는 네트웍을 통한 시간동기화 프로토콜이며, 로컬네트웍에서 마이크로초 (10^{-6}sec)이내의 정확도로 시간을 맞춘다.

PTP v1 & v2

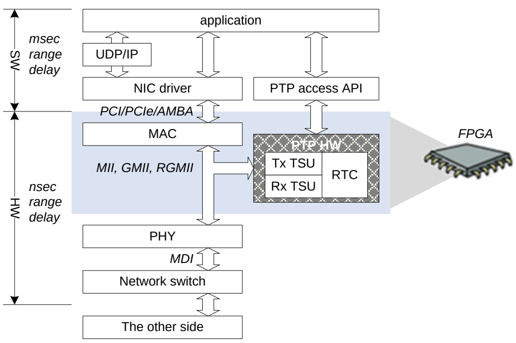
Criteria	PTPv1	PTPv2
clock types	Ordinary Clock (OC) Boundary Clock (BC)	Ordinary Clock (OC) Boundary Clock (BC) end-to-end Transparent Clock (e2e TC) peer-to-peer Transparent Clock (p2p TC) Management Node
time representation	epoc number (16 bit) seconds (32 bit) nanoseconds (32 bit)	seconds (48 bit) nanoseconds (32 bit)
time interval resolution	1 ns	2^{-16} ns (15.26 fs)
message types	Sync Follow_Up	Announce Sync Follow_Up
	Delay_Req Delay_Resp	Delay_Req Delay_Resp
	Management	Management
		Pdelay_Req Pdelay_Resp Pdelay_Resp_Follow_Up
		Signaling
message rates	small choice	bigger range and selectable per message type
addressing	multicast	multicast unicast
mappings	UDP/IPv4 over IEEE 802.3	UDP/IPv4 over IEEE 802.3 UDP/IPv6 over IEEE 802.3 directly over IEEE 802.3 PROFINET DeviceNet/ControlNet
extensions	none	by Type/Length/Value (TLV)
redundancy	BMC	BMC, Alternate Master, Master Cluster
	no	yes
multiple domains	by 4 multicast addresses	by Domain Number (8 bit)
What else?		profiles
		unicast message negotiation
		security protocol (experimental)

- V1 and V2 cannot directly synchronize to each other because they use different message format.
- V2
  - Transparent clock
  - More accurate clock (48-bit sec & 32-bit nsec)
    - Starts 1970.Jan.1: echo == 0
    - 48-bit second: up to 8_925_512 years
  - Peer-to-peer messages
  - UDP/IPv6 (layer 3) & Raw Ethernet (layer 2) packet
- Standard protocol only, but nothing about how to correct the slave.

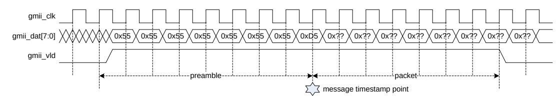
Hans Weibel, Technology Update on IEEE 1599: The Second Edition of the High Precision Clock Synchronization Protocol, 2009.

PTP implementation (example)

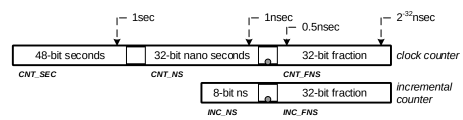
- PTP components
  - RTC: real-time clock
  - TSU: time-stamp unit
  - PTP API



- TSU takes time-stamp at the network side as close as possible in order to minimize time-varying.
  - Message time stamping at the end of preamble.



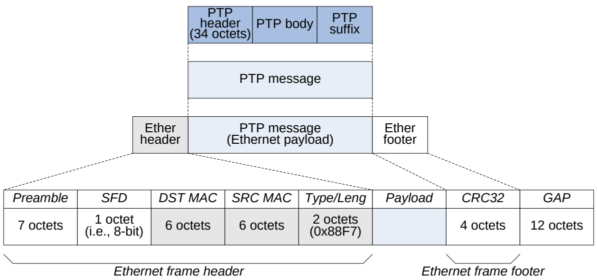
- RTC maintains two groups of counters; one is 'clock counter' and the other is 'incremental counter'



PTPv2 messages (1/3)

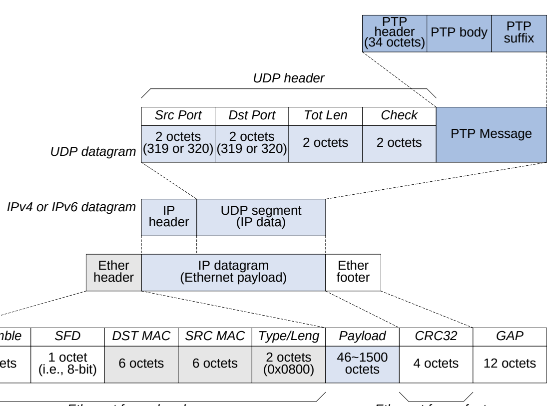
Message type	Message	Note
Event message	Sync	master to slave (to adjust offset)
	Delay_Req	slave to master (to measure propagation delay)
	Pdelay_Req	port to port (to measure link delay)
	Pdelay_Resp	port to port (to measure link delay)
General message	Announce	
	Follow_Up	master to slave (to adjust offset)
	Delay_Resp	master to slave (to measure propagation delay)
	Pdelay_Resp_Follow_Up	port to port (to measure link delay)
	Management	
	Signaling	

PTPv2 messages (2/3)



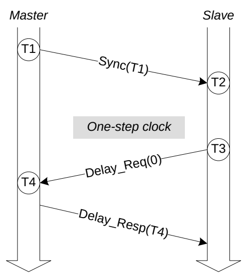
- Raw Ethernet
  - Ref: IEEE 1588-2008 Annex F
  - Ethernet type field: 0x88F7.
  - Destination MAC address
    - 01-1B-19-00-00-00 (delay message)
      - Sync, Announce
    - 01-80-C2-00-00-0E (peer delay message)
      - Pdelay_Req

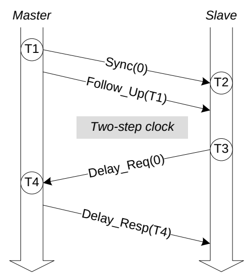
PTPv2 messages (3/3)



- UDP/IP over Ethernet
  - Ref: IEEE 1588-2008 Annex D and E
  - Ethernet type field: 0x0800 (IP)
  - UDP port: 319 or 320
    - 319 for even messages: Sync, Delay_Req
    - 320 for general message: Follow_Up, Delay_Resp
  - Destination IP address (multicast)
    - IPv4
      - 224.0.1.129 (delay message) 0xE0.00.01.81
        - MAC: 01-00-5E-00-01-81
      - 224.0.0.107 (peer delay message) 0xE0.00.00.6B
        - MAC: 01-00-5E-00-00-6B
    - IPv6
      - FF0x::181 (delay message)
      - FF02::6B (peer delay message)

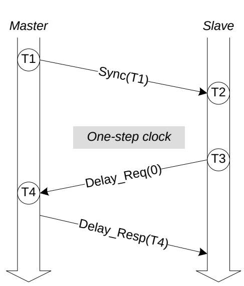
Mechanisms to measure offset

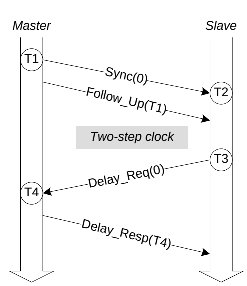




- Under the assumption of symmetric link
  - propagation time from master to slave equals to the propagation time from slave to master
  - offset = (Slave Time) – (Master Time) = [(T2-T1)-(T4-T3)]/2
- For **one-step**, the 'Sync' message carries time-stamp, while 'Follow_Up' carries time-stamp for **two-step**.

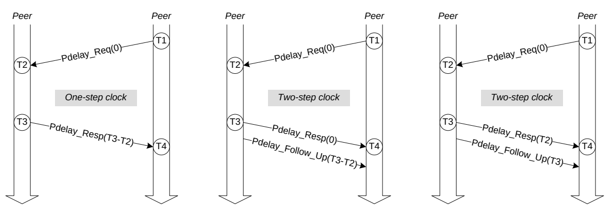
Mechanisms to measure delay (1/2)





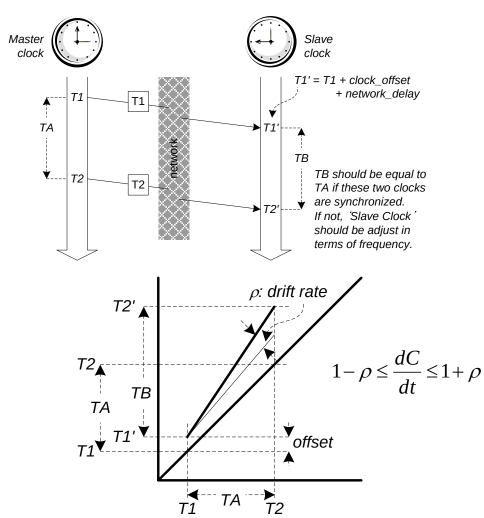
- Delay request-response mechanism
  - to generate and communicate the timing information needed to synchronize ordinary and boundary clocks
  - 1-step clock
    - Sync, Delay_Req, Delay_Resp
    - [(T2-T1)+(T4-T3)]/2
  - 2-step clock
    - Sync, ***Follow_Up***, Delay_Req, Delay_Resp
- For **one-step**, the 'Sync' message carries time-stamp, while 'Follow_Up' carries time-stamp for **two-step**.

Mechanisms to measure delay (2/2)



- Peer delay mechanism
  - to <u>measure the link delay between two clock ports</u>
  - 1-step clock
    - Pdelay_Req, Pdelay_Resp
    - [(T4-T1)-(T3-T2)]/2
  - 2-step clock
    - Pdelay_Req, Pdelay_Resp, ***<u>Pdelay_Resp_Follow_Up</u>***
    - [(T2-T1)+(T4-T3)]/2

Mechanisms to measure drift

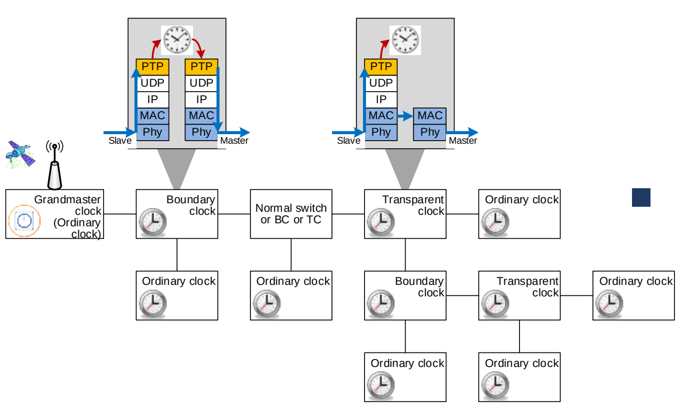


- Drift_Rate
  - [(T2'-T1')/(T2-T1)] – 1
  - TB/TA – 1 = (TB – TA) / TA
- If Slave clock increments its frequency with Fs, it should be adjusted as follows.
  - Fs' = Fs – (Fs * Drift_Rate)

PTP clocks (1/2)

PTP devices (clock and node)	Remark
OC: Ordinary clock	A **single port device** that can be a Master or Slave clock.
GMC: grandmaster clock	
SC: Slave clock	
CC: Connection clock	A **multi port device**
BC: Boundary clock	A **multi port device** that can be a Master or Slave clock. It has a **built-in Slave** clock that recovers a clock. This clock is then used to drive the **built-in Master**, which supplies the clock to the next node.
TC: Transparent clock	A multi-port device that is not a master or slave clock but a bridge between the two.
E2E TC: End-to-End TC	**Forwards and modifies all PTP Messages** to compensate for residence time. Compensation is achieved by addition of the bridge residence time into a correction field within the header of the message.
P2P TC: Peer-to-Peer TC	**Forwards and modifies Sync and Follow_Up messages** only to compensate for residence time. Compensation is achieved by addition of the bridge residence time + the peer-to-peer link delay, into a correction field within the header of the message.
Management node	A device that configures and monitors clocks.

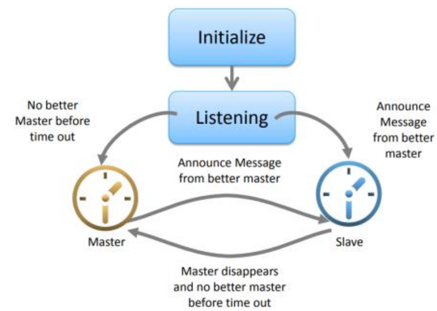
PTP clocks (2/2)



- BC (Boundary Clock) extends synchronization across an intermediate network element
  - A boundary clock contains more than one PTP port:
    - a slave port that is synchronized with a remote master, and
    - a master port that synchronizes other slaves downstream
  - Synchronization messages are terminated at each port and not forwarded
- TC (Transparent Clock) is a switch that adjusts a PTP message's timestamp to compensate for its own queuing delays
  - Timestamp in incoming message is modified before sending the message out
    - Creates security issues, since original crypto checksum is not valid anymore

PTP best master clock algorithm

- The *best master clock* (BMC) algorithm performs a distributed selection of the best candidate clock based on the clock properties.
  - Any ordinary clock can be grand master clock.
- Clock properties are broadcasted via 'Sync' (PTPv1) or 'Announce' (PTPv2) message. (lower has higher priority)
  - 1. priority (0-255): user configurable
  - 2. class (0-255): got GPS
  - 3. accuracy: 100ns
  - 4. variance: frequency stability
  - 5. priority (0-255): user configurable
  - 6. unique identifier (clock id)



PTP profiles

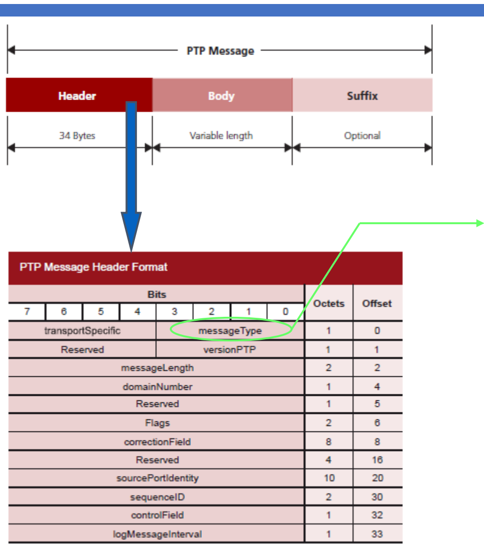
- The set of allowed PTP features applicable to a device. (more restrictive set of rules for specific application)
  - Specifies
    - Required options
    - Allowed options
    - Forbidden options
    - Network topology limitations
    - Performance requirements
- Default profile
  - Defined by IEEE 1588 (it only defines this default profile)
- Power plant profile
  - IEEE C37.238-2011/2017 Standard
- Telecom profile for frequency transfer
  - ITU-T G.8265, G.8275
- Audio & video application profile
  - IEEE 802.1AS
- Enterprise profile

Profiles are a set of rules which place restrictions on PTP, intended to meet the needs of a specific application or set of similar applications.

Table of Contents

- Background
- IEEE 1588 PTP
- IEEE 1588 PTPv2 messages

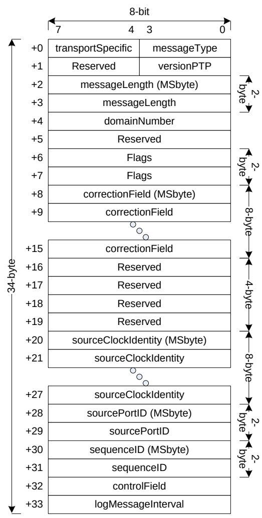
PTPv2 message



- All PTP message consists of
  - header (34-byte)
    - common to all PTP message
  - body (variable length)
  - suffix (optional)

Type	code	Message
Event message	0x0	Sync
	0x1	Delay_Req
	0x2	Pdelay_Req
	0x3	Pdelay_Resp
General message	0x8	Follow_Up
	0x9	Delay_Resp
	0xA	Pdelay_Resp_Follow_Up
	0xB	Announce
	0xC	Signaling
	0xD	Management

PTPv2 message header



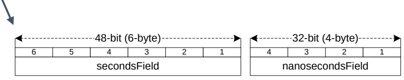
- The header (34-byte) is common to all PTP message. (note that big-endian style for multi-byte data)
  - transportSpecific: ‘0’ or ‘1’ depending on UDP payload 124-byte requirement
    - 1 for IEEE 802.1AS and 0 by default
  - messageType: Sync, Delay_Req, …
  - versionPTP: PTP version of the originating node
    - 2 for PTPv2
  - messageLength: full length including header, body and suffix, but excluding any padding
  - domainNumber: domain this message belongs to
    - domain: logical group of clocks to be synchronized
  - flags: status information
  - correctionField: correction value in nano-second
  - souceClockIdentity
  - sourcePortIdentity: originating port for this message
  - sequenceID: sequence number of each message types
  - controlField: backward compatibility
  - logMessageInterval: depends on type of message

Announce message

- Indicates the capabilities of a clock to the other clocks on the same domain.

Announce Message Format

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (13.3)	34	0
originTimestamp	10	34
currentUtcOffset	2	44
Reserved	1	46
grandmasterPriority1	1	47
grandmasterClockQuality	4	48
grandmasterPriority2	1	52
grandmasterIdentity	8	53
stepsRemoved	2	61
timeSource	1	63



Sync & Follow_Up messages

Sync Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
originTimestamp								10	34

48-bit (6-byte)						32-bit (4-byte)			
6	5	4	3	2	1	4	3	2	1
secondsField						nanosecondsField			

- The ‘Sync’ message is sent by a master clock and contains the master time
  - If the master clock is a two-step, this timestamp will be zero value. Actual timestamp value will be sent afterwards in the associated ‘Follow_Up’ message.

Follow_Up Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
preciseOriginTimestamp								10	34

48-bit (6-byte)						32-bit (4-byte)			
6	5	4	3	2	1	4	3	2	1
secondsField						nanosecondsField			

- The ‘Follow_Up’ message is sent by master clock and contains the master time
  - It is used when the master clock is a two-step.

Delay_Req & Delay_Resp messages

Delay_Req Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
originTimestamp								10	34

48-bit (6-byte)						32-bit (4-byte)			
6	5	4	3	2	1	4	3	2	1
secondsField						nanosecondsField			

- The ‘Delay_Req’ message is sent by a slave clock and contains the slave time
  - It is only used in the delay request-response mechanism.

Delay_Resp Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
receiveTimestamp								10	34
requestingPortIdentity								10	44

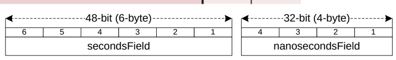
48-bit (6-byte)						32-bit (4-byte)			
6	5	4	3	2	1	4	3	2	1
secondsField						nanosecondsField			

- The ‘Delay_Resp’ message is sent by the master clock and contains the master time when the ‘Delay_Req’ message was received.
  - It is only used in the delay request-response mechanism.
  - ‘requestingPortIdentity’ contains the value of ‘sourcePortIdentity’ field of the associated ‘Delay_Req’ message.

Pdelay_Req & Pdelay_Resp messages

Pdelay_Req Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
originTimestamp								10	34
reserved								10	44



- The ‘Pdelay_Req’ message is sent by ‘delay requester’ peer-to-peer clock and contains the ‘delay requester’ peer-to-peer clock time.
  - It is sent only in the peer delay mechanism.
  - ‘reserved’ filed is used to make the message length the same as the ‘Pdelay_Resp’ message.

Pdelay_Req & Pdelay_Resp messages

Pdelay_Resp Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
receiveReceiptTimestamp								10	34
requestingPortIdentity								10	44

Pdelay_Resp_Follow_Up Message Format

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
header (13.3)								34	0
responseOriginTimestamp								10	34
requestingPortIdentity								10	44

- The ‘Pdelay_Resp’ message is sent by ‘delay responder’ peer-to-peer clock and contains the ‘delay responder’ peer-to-peer clock time.
  - It is sent only in the peer delay mechanism.
  - ‘requestingPortIdentity’ contains the value of ‘sourcePortIdentity’ field of the associated ‘Pdelay_Req’ message.
  - If the clock is a two-step, this timestamp will be zero value. Actual timestamp value will be sent afterwards in the associated ‘Pdelay_Resp_Follow_Up’ message.
- The ‘Pdelay_Resp_Follow_Up’ message is sent by ‘delay responder’ peer-to-peer clock.

Signaling and management messages

Signalling Message Format

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (13.3)	34	0
targetPortIdentity	10	34
One or more TLVs	N	44

Management Message Format

Bits (7 6 5 4)	Bits (3 2 1 0)	Octets	Offset
header (13.3)		34	0
targetPortIdentity		10	34
startingBoundaryHops		1	44
boundaryHops		1	45
Reserved	actionField	1	46
Reserved		1	47
managementTLV		M	48

Bits (7 6 5 4 3 2 1 0)	Octets	TVL Offset
tlvType	2	0
lengthField	2	2
managementID	2	4
dataField	N	6

- Signaling message
  - Send TLV (Type Length Value)
    - REQUEST_UNICAST_TRANSMISSION
    - GRANT_UNICAST_TRANSMISSION
    - CANCEL_UNICAST_TRANSMISSION
    - ...
- Management message
  - to transmit information from a clock to a node manager and from a node manager to one or more clocks.

References

- IEEE 1588-2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.
- RFC 5905, Network Time Protocol Version 4: Protocol and Algorithms Specification, 2010.6.
- RFC 2030, Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI, 1996.10.
- J.C. Eidson, Measurement, Control and Communication Using IEEE 1588, Springer, 2010.
- D.L. Mills, Computer Network Time Synchronization, 2nd Edition, CRC Press, 2010.

㈜퓨쳐디자인시스템
34051 대전광역시 유성구 문지로 193, KAIST 문지캠퍼스, F723호
(042) 864-0211~0212 / contact@future-ds.com / www.future-ds.com

Future Design Systems, Inc.
Faculty Wing F723, KAIST Munji Campus, 193 Munji-ro, Yuseong-gu, Daejeon 34051, Korea
+82-042-864-0211~0212 / contact@future-ds.com / www.future-ds.com

FUTURE
Design Systems