OCP-TAP

LET'S TALK SERVOS

TIME APPLIANCES

17 JANUARY 2024
GREG ARMSTRONG
PRINCIPAL SYSTEM ARCHITECT, TIMING
ANALOG & CONNECTIVITY PRODUCT GROUP
RENESAS ELECTRONICS

RENESAS

INTRODUCTION

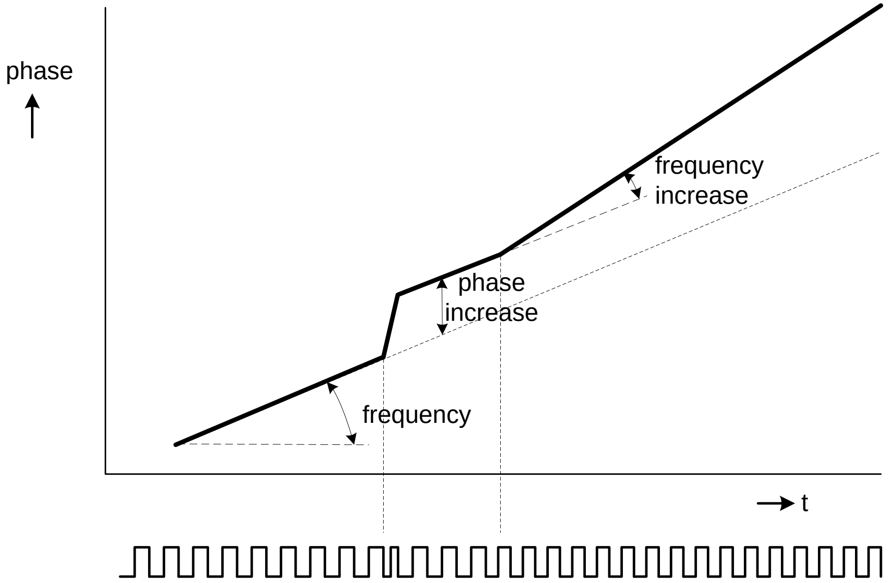
- Clock synchronization is essential in numerous applications, ranging from telecommunications and network protocols to industrial automation and financial transactions
- Accuracy of synchronized clocks is paramount, as deviations in timekeeping can lead to:
  - data inconsistencies
  - communication errors
  - operational disruptions
- Servos (servomechanisms) play a critical role in achieving accurate clock synchronization

TIME, PHASE & FREQUENCY

We all know what time is - unit is second [s]

Phase is the angle of a rotating vector; in clock terms: time related to the period of a repetitive signal - unit is radians [rad]

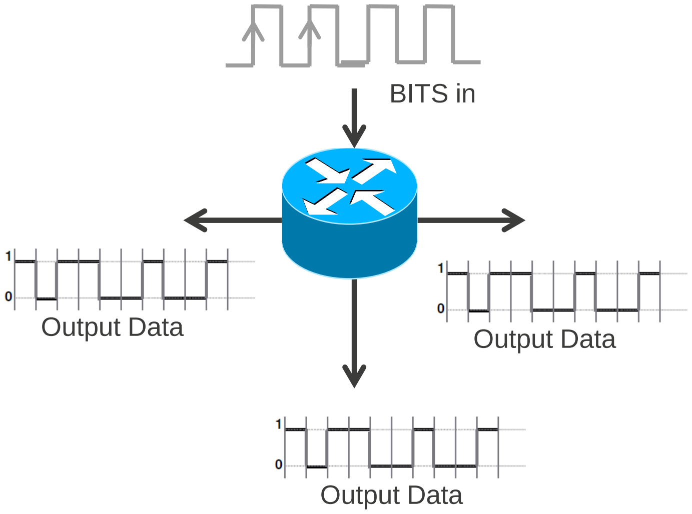
Frequency is a statistical term; number of events per second - unit is Hertz [Hz]



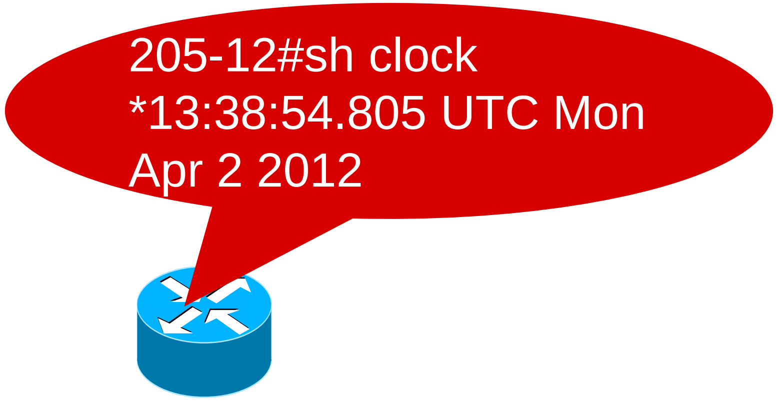
THERE ARE TWO “THINGS” TO SYNCHRONIZE

- Frequency and Time

Frequency



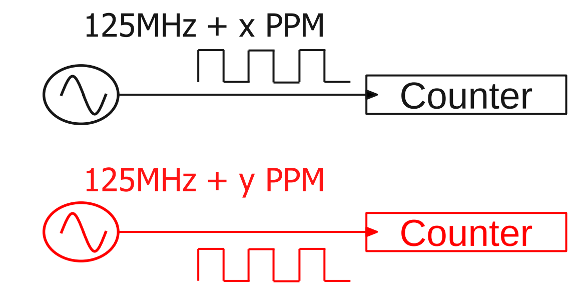
Time/Phase

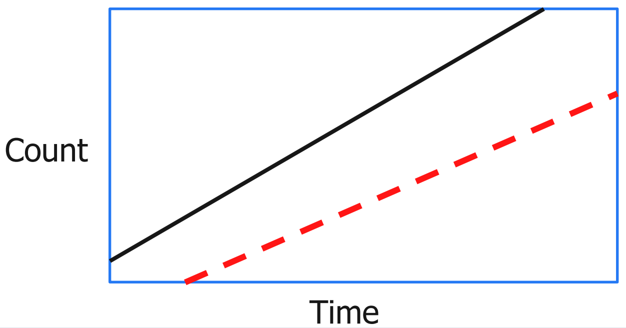


```
205-12#sh clock
*13:38:54.805 UTC Mon
Apr 2 2012
```

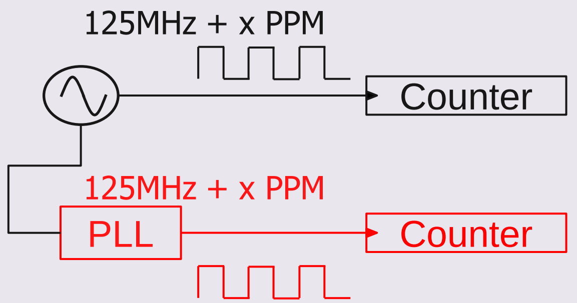
BITS … Building Integrated Timing Supply

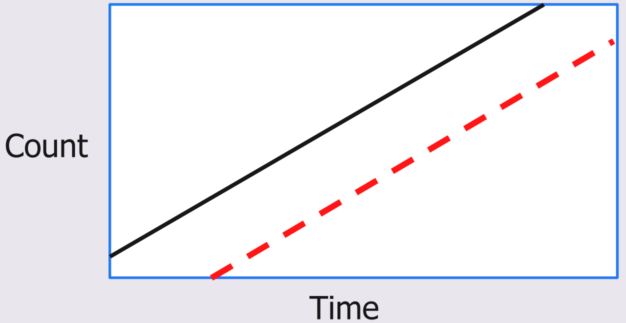
WHAT IS “SYNCHRONIZATION”



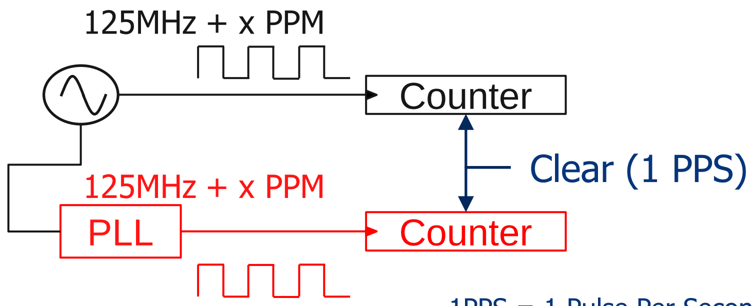


Not Synchronized

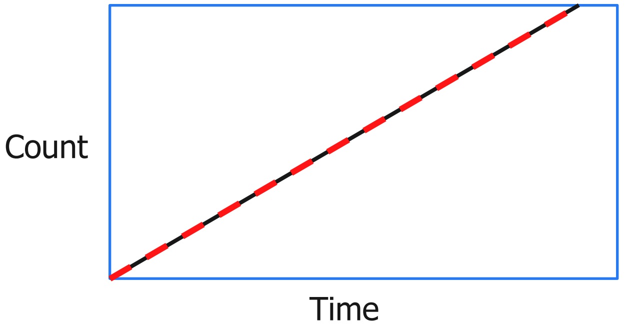




Frequency Synchronized (Syntonised)



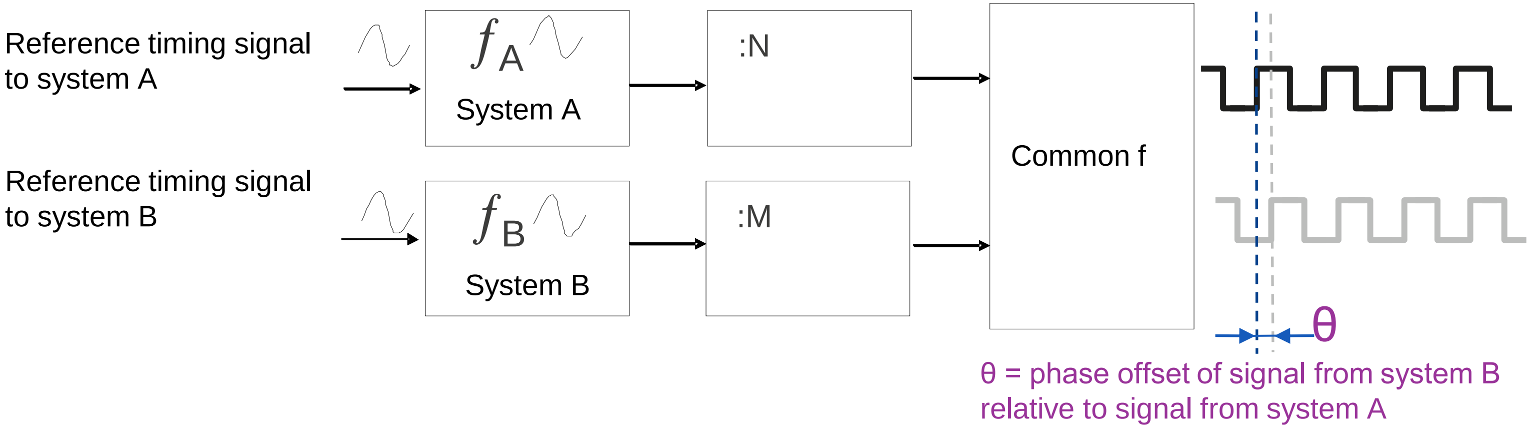
1PPS = 1 Pulse Per Second



Phase/Time Synchronized

FREQUENCY SYNCHRONIZATION

FREQUENCY SYNCHRONIZATION


Reference timing signal to system A
f_A
System A
:N
Reference timing signal to system B
f_B
System B
:M
Common f
θ
θ = phase offset of signal from system B relative to signal from system A

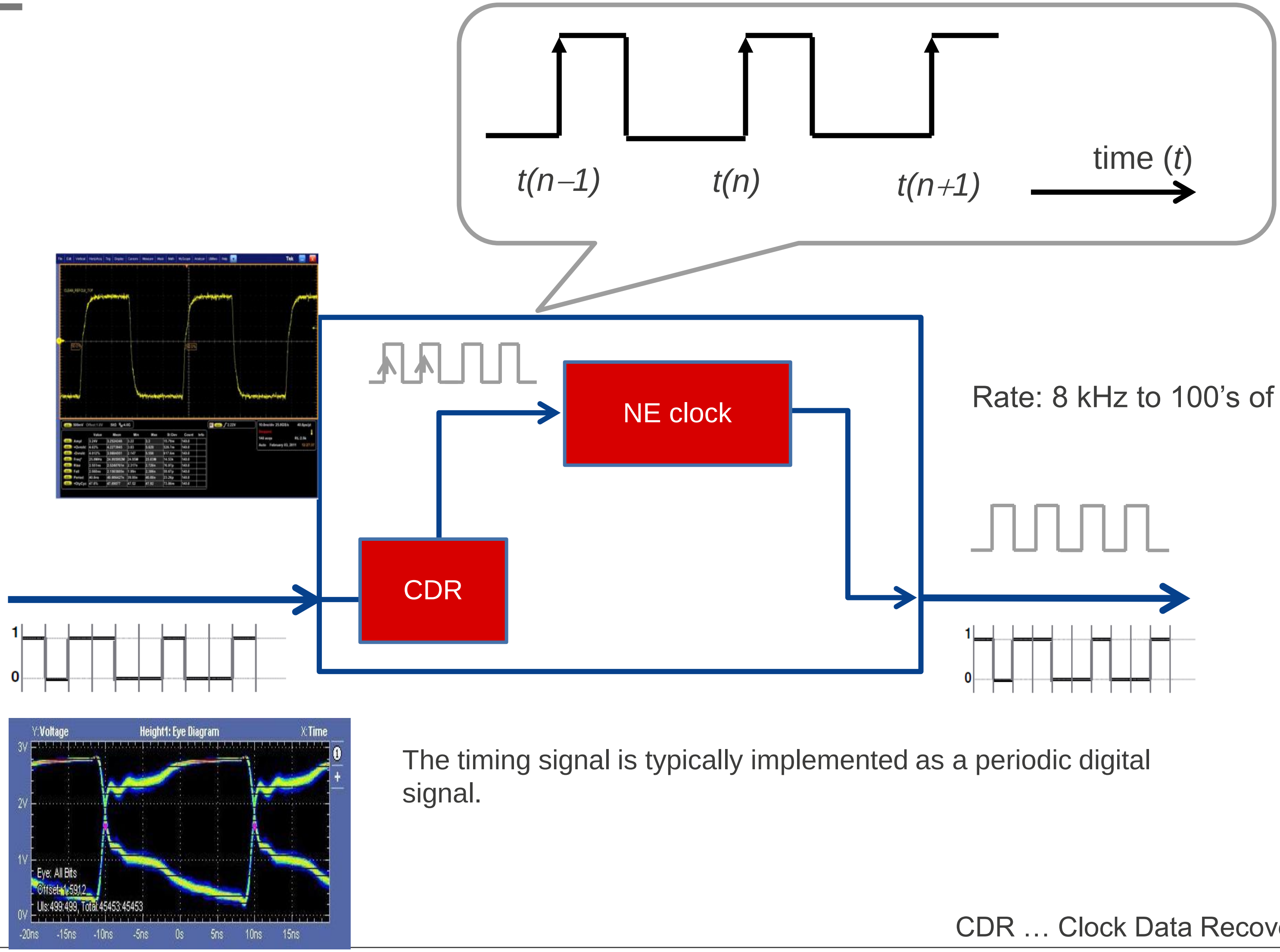
Two clocks are frequency synchronized if the frequency of the two clocks have common denominator.

Two clocks are called plesiochronous if the difference in their common denominator is bounded.

The difference in position of rising edges of the clocks is called phase offset.

Two common frequencies which have constant phase offset are phase-locked and implicitly frequency synchronized.

PHYSICAL LAYER FREQUENCY DISTRIBUTION



Rate: 8 kHz to 100’s of MHz

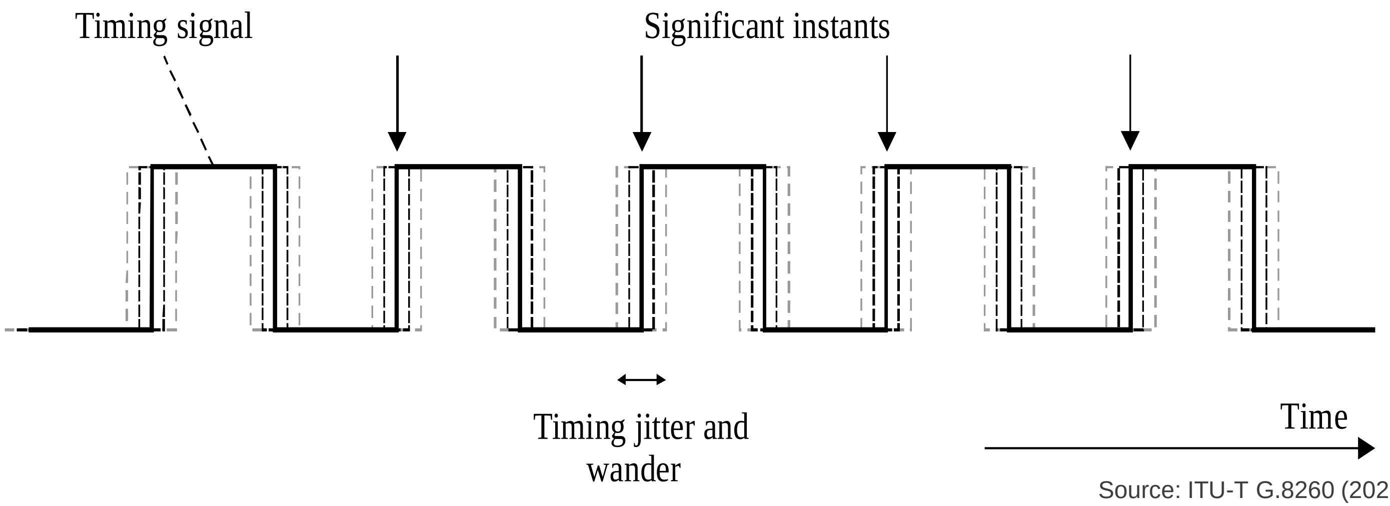
The timing signal is typically implemented as a periodic digital signal.

CDR … Clock Data Recovery; NE … Network Element

SIGNIFICANT INSTANTS – PHYSICAL DISTRIBUTION

- Timing Signal and Noise

Rate: 8 kHz to 100’s of MHz



Timing signal

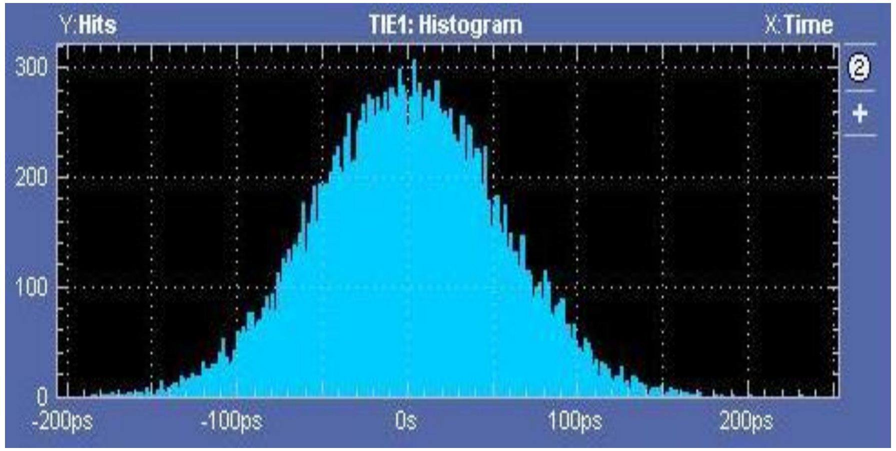
Significant instants

Timing jitter and wander

Time

Source: ITU-T G.8260 (202211)

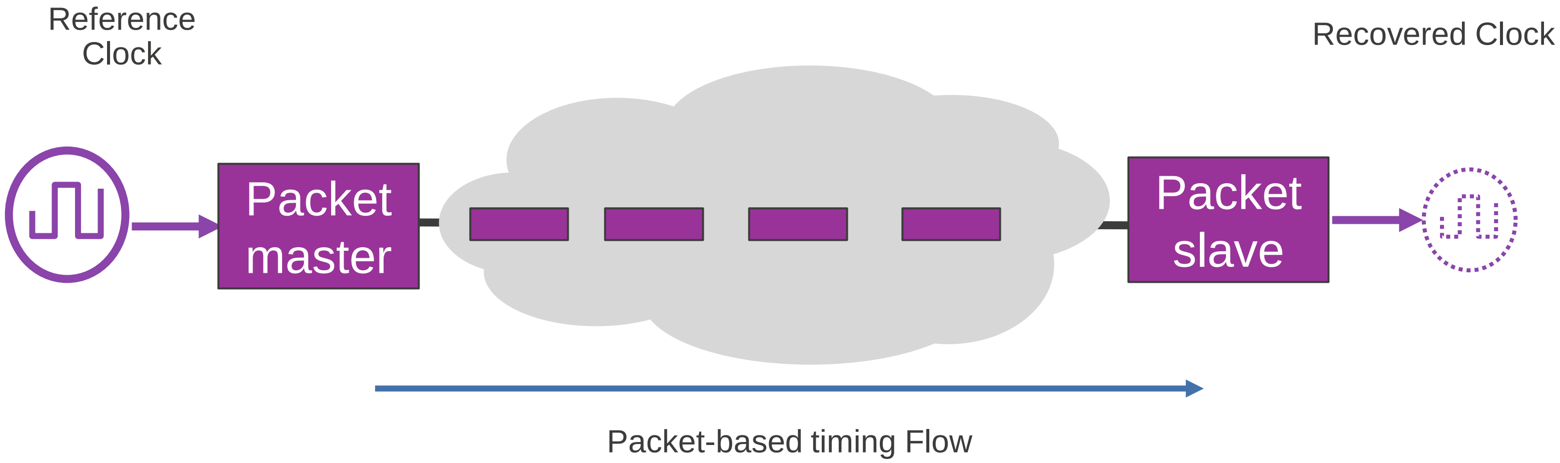
Example : 25 MHz signal



TIE … Time Interval Error

PACKET-BASED FREQUENCY DISTRIBUTION

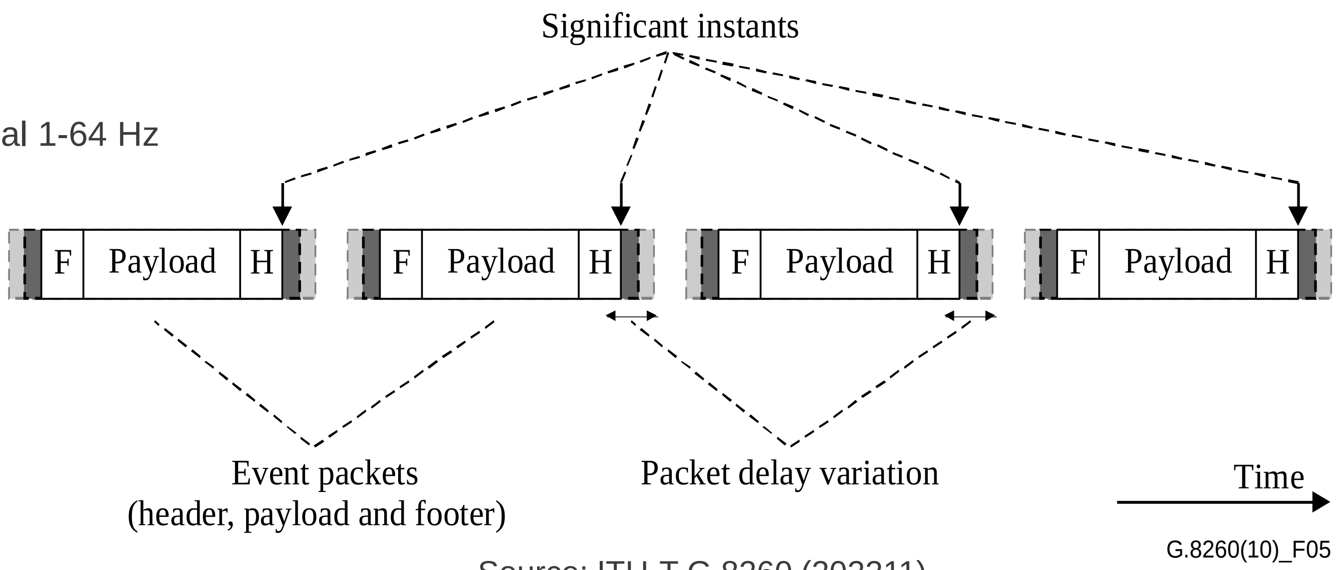
- Three key steps:
  - **Generation**: from physical signal to packet
  - **Transfer**: timing events (frame or packet flow) transmission over packet network
  - **Recovery**: from packet-based signal to physical signal



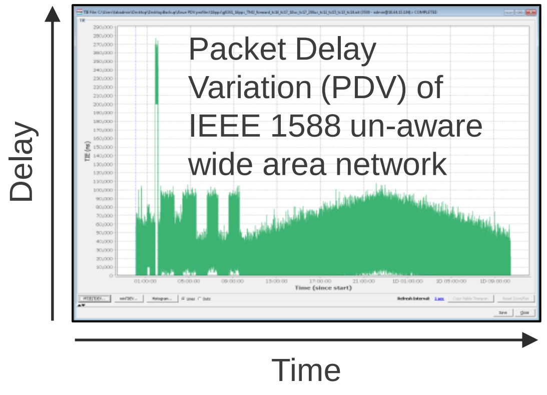
SIGNIFICANT INSTANTS PACKET-BASED DISTRIBUTION

- Timing signal could either be
  - Periodic with known send frequency (e.g., CES) or
  - With additional information (e.g., **PTP timestamps**) defining the ideal position in time of the significant instant relative to a master time scale.

Rate : typical 1-64 Hz



Source: ITU-T G.8260 (202211)



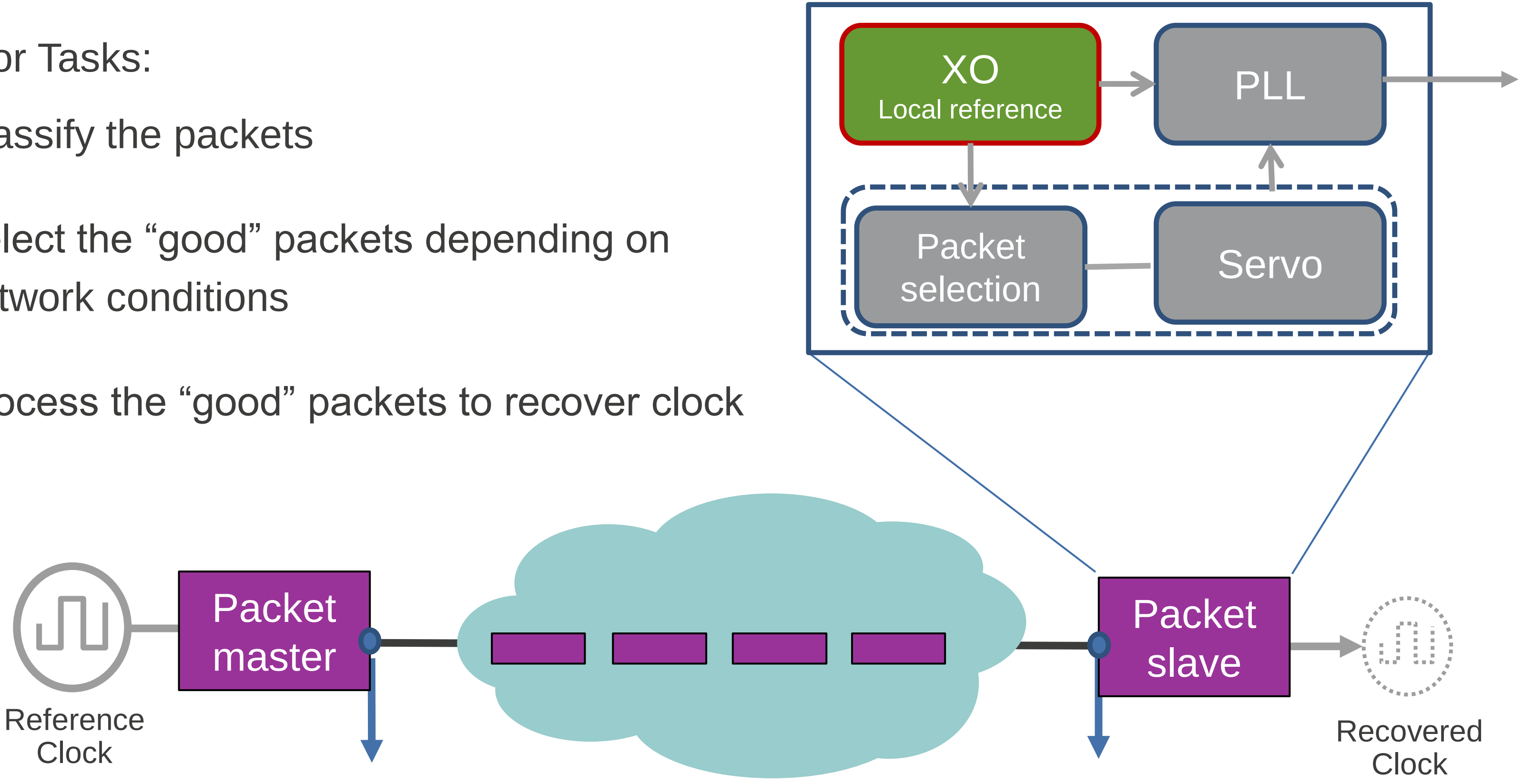
PDV profiles per ITU-T G.8261 Appendix VI

CES … Circuit Emulation Service; PTP … Precision Time Protocol

PACKET-BASED TIMING RECOVERY

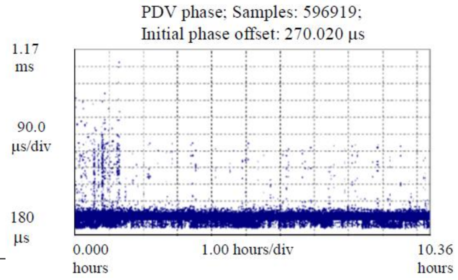
Three Major Tasks:

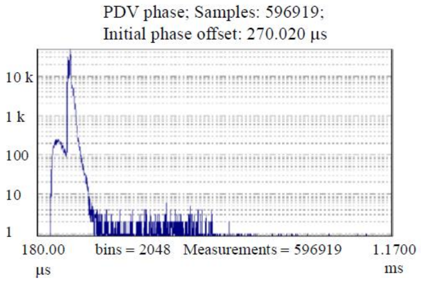
1. Classify the packets
2. Select the “good” packets depending on network conditions
3. Process the “good” packets to recover clock



Packet-based timing Flow

Source: ITU-T G.8260 (202211)





XO … Oscillator; PLL … Phase Locked Loop; PDV … Packet Delay Variation

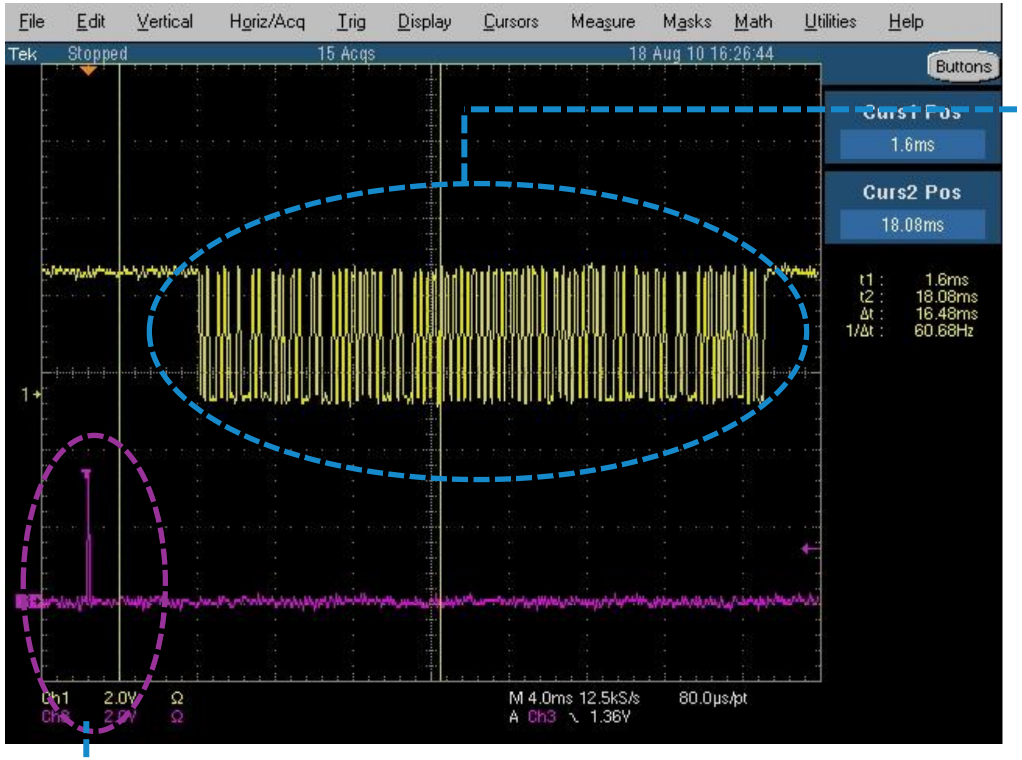
TIME SYNCHRONIZATION

TIME – A CLOSER LOOK

- Time = Phase + Time of Day

*13:38:54.805 UTC Mon Apr 2 2012

Time of Day (TOD) Information
(serial interface on the RJ45 connector)

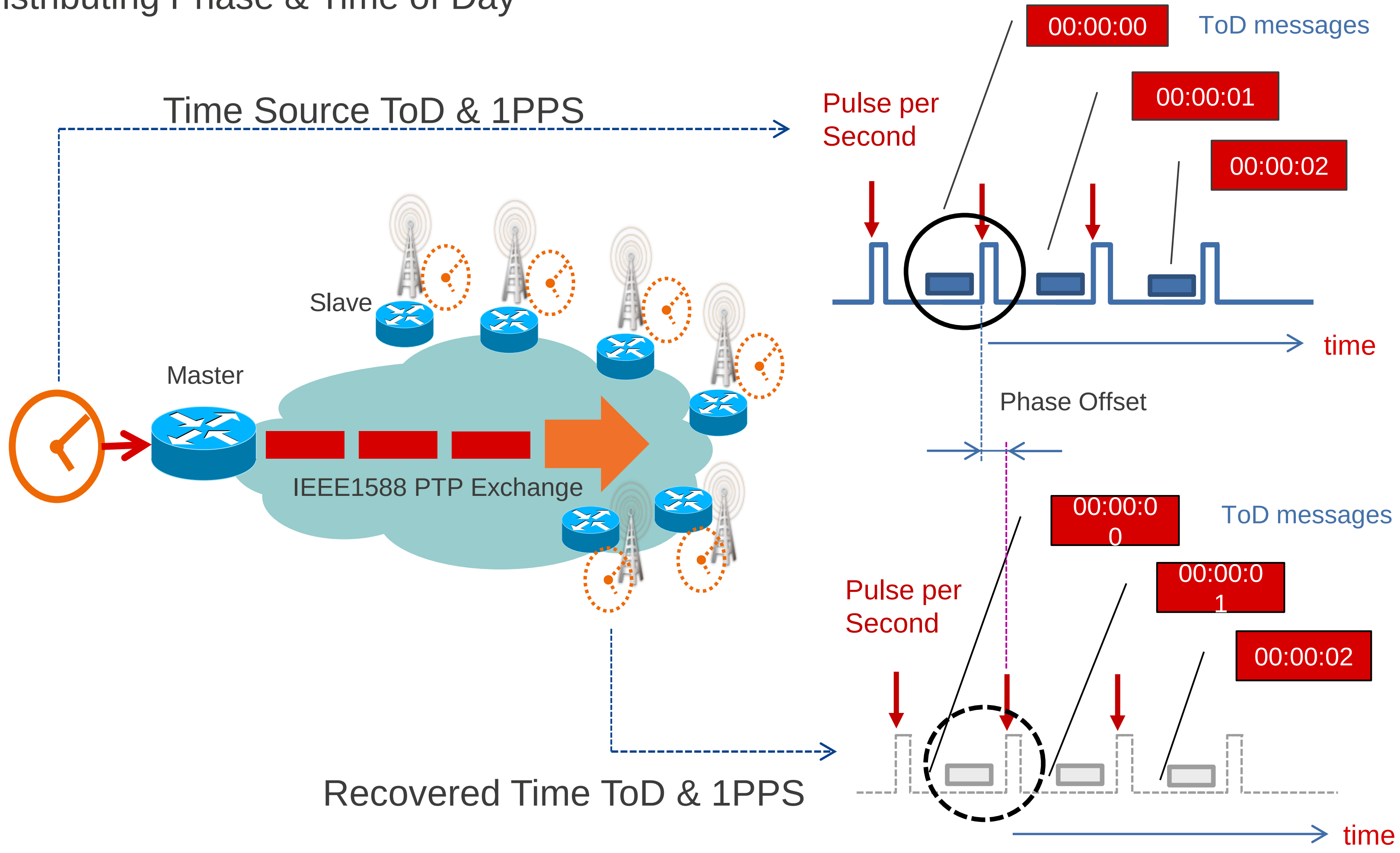


```
Tek   Stopped          15 Acqs               18 Aug 10 16:26:44

Curs1 Pos  1.6ms
Curs2 Pos  18.08ms

t1 :    1.6ms
t2 :    18.08ms
Δt :    16.48ms
1/Δt :  60.68Hz

Ch1  2.0V  Ω                 M 4.0ms 12.5kS/s    80.0μs/pt
Ch3  2.0V  Ω                 A  Ch3  ↘  1.36V
```

1PPS Pulse → Phase
(analog signal on the DIN connector)

TIME DISTRIBUTION

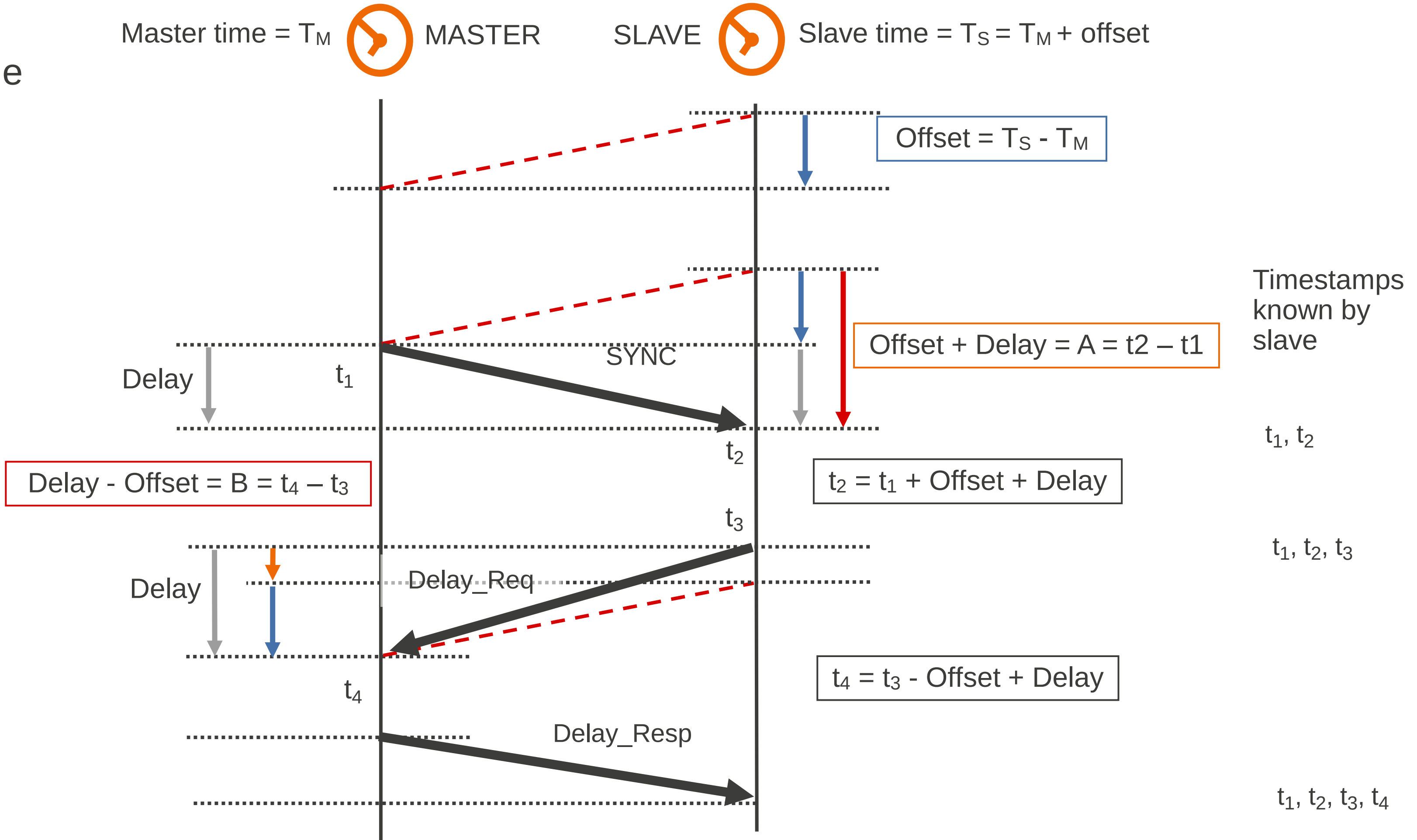
- Packet Layer distributing Phase & Time of Day



SYNCHRONIZATION PROTOCOLS

TWTT PROTOCOL BASICS

- Basic PTP Message Exchange

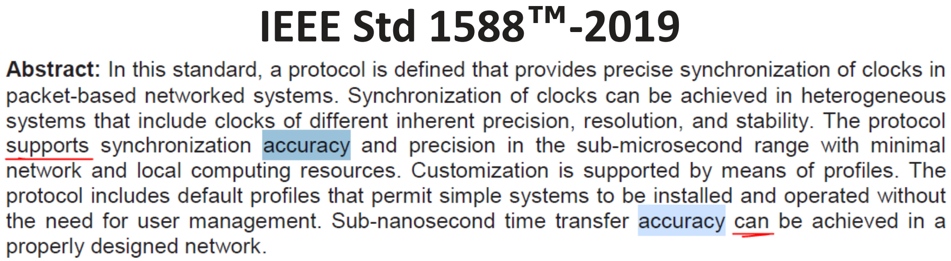

Master time = T_M
MASTER
SLAVE
Slave time = $T_S = T_M$ + offset
Offset = $T_S - T_M$
Timestamps known by slave
Offset + Delay = A = t2 – t1
SYNC
Delay
t_1
t_2
t_1, t_2
Delay - Offset = B = $t_4 - t_3$
$t_2 = t_1$ + Offset + Delay
t_3
t_1, t_2, t_3
Delay
Delay_Req
t_4
$t_4 = t_3$ - Offset + Delay
Delay_Resp
t_1, t_2, t_3, t_4

Delay = $((t_2 - t_1)+(t_4 - t_3))/2$

Offset = $((t_2 - t_1)-(t_4 - t_3))/2$

TWTT … Two-Way Time Transfer

WHAT IS PRECISION TIME PROTOCOL (PTP)?

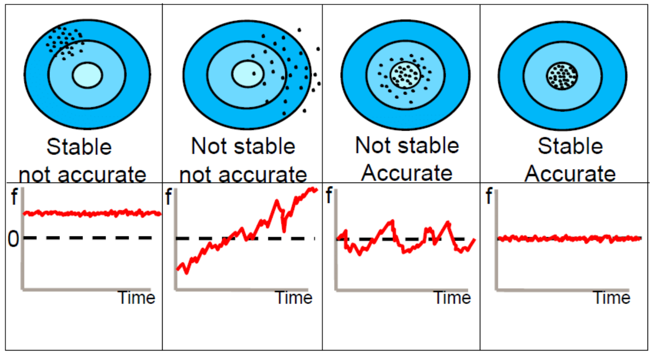


IEEE Std 1588™-2019

Abstract: In this standard, a protocol is defined that provides precise synchronization of clocks in packet-based networked systems. Synchronization of clocks can be achieved in heterogeneous systems that include clocks of different inherent precision, resolution, and stability. The protocol supports synchronization accuracy and precision in the sub-microsecond range with minimal network and local computing resources. Customization is supported by means of profiles. The protocol includes default profiles that permit simple systems to be installed and operated without the need for user management. Sub-nanosecond time transfer accuracy can be achieved in a properly designed network.

E.g., Uncompensated XTAL/MEMS

E.g., Precision Servo

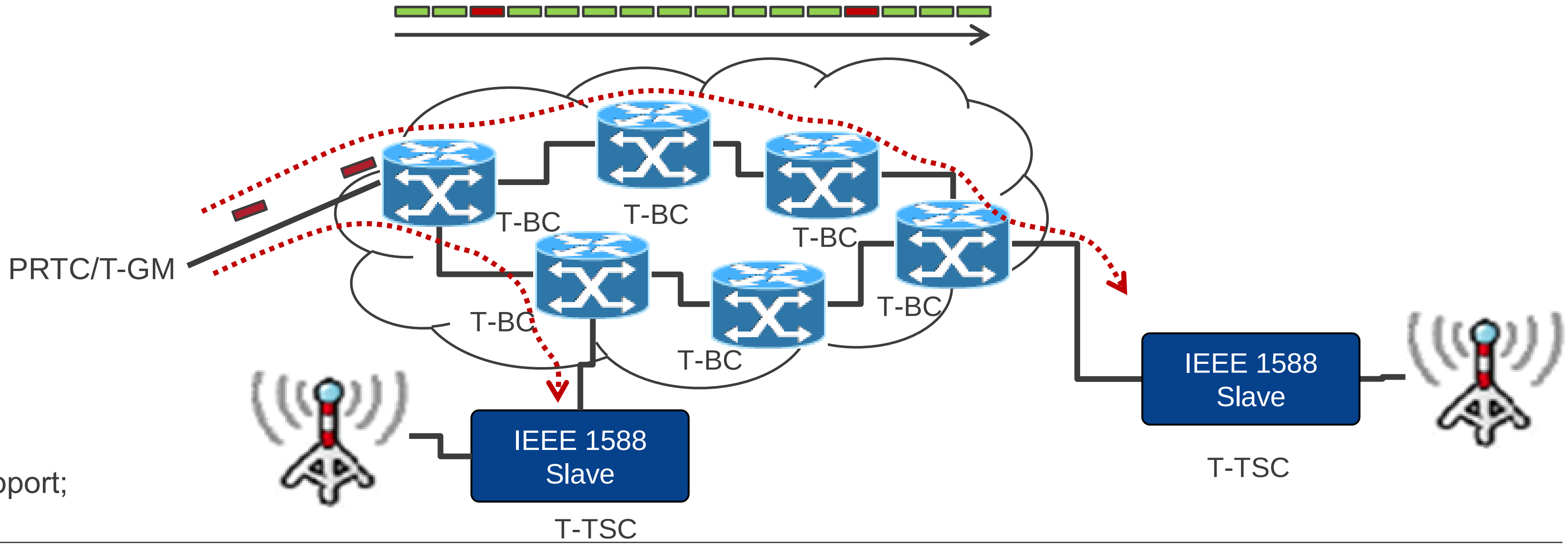


E.g., High-Stability XO

E.g., Precision Measurement

Protocol defined by the IEEE in standard 1588-2008 (v2) or 1588-2019 (v2.1)

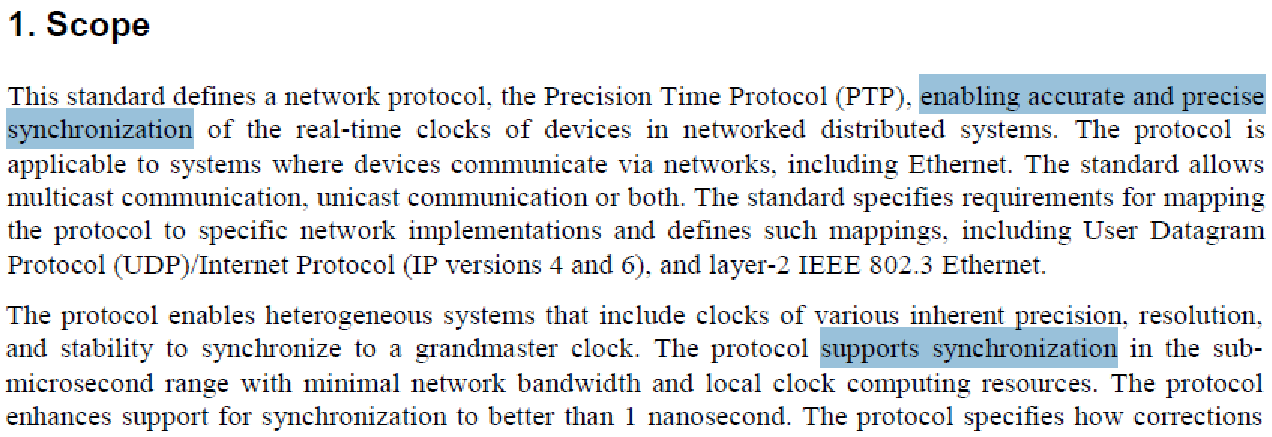
- Profile (how to configure the protocol and what optional features to use, if applicable) defined by the ITU-T in recommendation G.8265.1 (SOOC), G.8275.1 (FTS) or G.8275.2 (APTS/PTS)
  - Other industries define their own profiles
- An OC or BC with a PTP port in the SLAVE state must synchronize its local PTP clock to the GM clock – but means of synchronization is outside scope of IEEE standard (and sometimes not even mentioned in the Profile)



FTS … Full Time Support; PTS … Partial Time Support;
APTS … Assisted PTS

I REPEAT, 1558 IS A PROTOCOL

- The biggest misunderstand of what 1588 provides is time or clock “synchronization” – precision vs accuracy
- It’s a time/clock distribution protocol, but by itself, it only provides a measurement of time/phase offset at a specific instance of sending and receiving an event packet

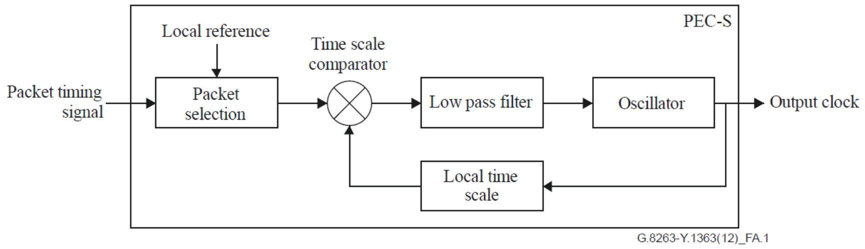


1. Scope

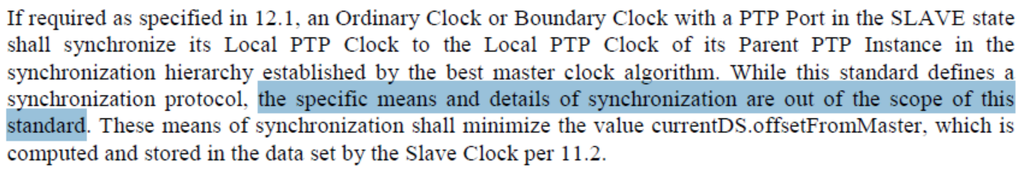
This standard defines a network protocol, the Precision Time Protocol (PTP), enabling accurate and precise synchronization of the real-time clocks of devices in networked distributed systems. The protocol is applicable to systems where devices communicate via networks, including Ethernet. The standard allows multicast communication, unicast communication or both. The standard specifies requirements for mapping the protocol to specific network implementations and defines such mappings, including User Datagram Protocol (UDP)/Internet Protocol (IP versions 4 and 6), and layer-2 IEEE 802.3 Ethernet.

The protocol enables heterogeneous systems that include clocks of various inherent precision, resolution, and stability to synchronize to a grandmaster clock. The protocol supports synchronization in the sub-microsecond range with minimal network bandwidth and local clock computing resources. The protocol enhances support for synchronization to better than 1 nanosecond. The protocol specifies how corrections

- Looking a tradition PLL model, all PTP provides is the function of the phase/frequency detector (PFD)



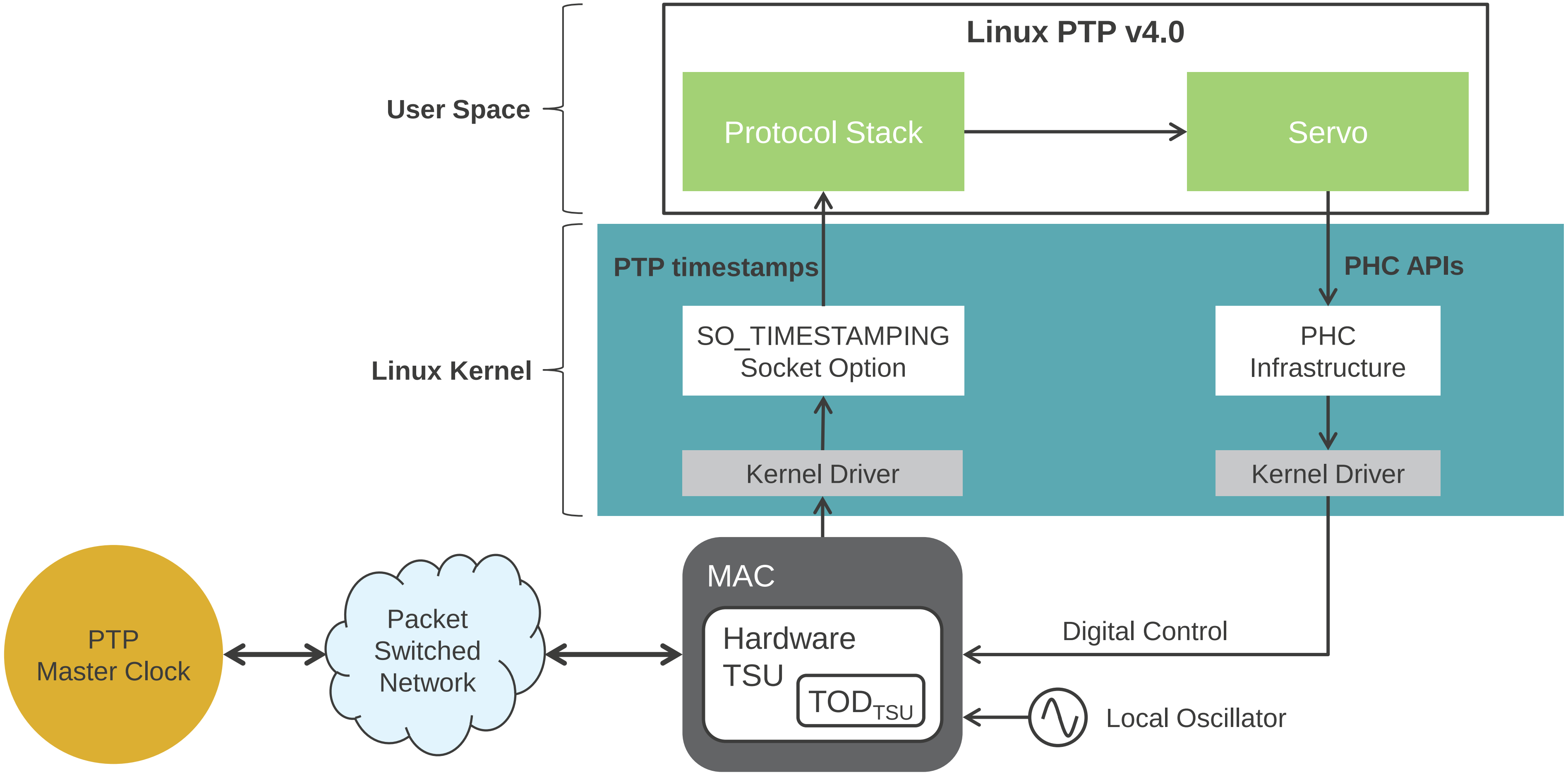
You still need a servo!



If required as specified in 12.1, an Ordinary Clock or Boundary Clock with a PTP Port in the SLAVE state shall synchronize its Local PTP Clock to the Local PTP Clock of its Parent PTP Instance in the synchronization hierarchy established by the best master clock algorithm. While this standard defines a synchronization protocol, the specific means and details of synchronization are out of the scope of this standard. These means of synchronization shall minimize the value currentDS.offsetFromMaster, which is computed and stored in the data set by the Slave Clock per 11.2.

IEEE Std 1588™-2019

TYPICAL LINUX PTP IMPLEMENTATION



PHASE-LOCKED LOOP (PLL)

… A CONTROL SYSTEM THAT GENERATES AN OUTPUT SIGNAL WHOSE PHASE IS RELATED TO THE PHASE OF AN INPUT SIGNAL. THERE ARE SEVERAL DIFFERENT TYPES; THE SIMPLEST IS AN ELECTRONIC CIRCUIT CONSISTING OF A VARIABLE FREQUENCY OSCILLATOR AND A PHASE DETECTOR IN A FEEDBACK LOOP. WIKIPEDIA

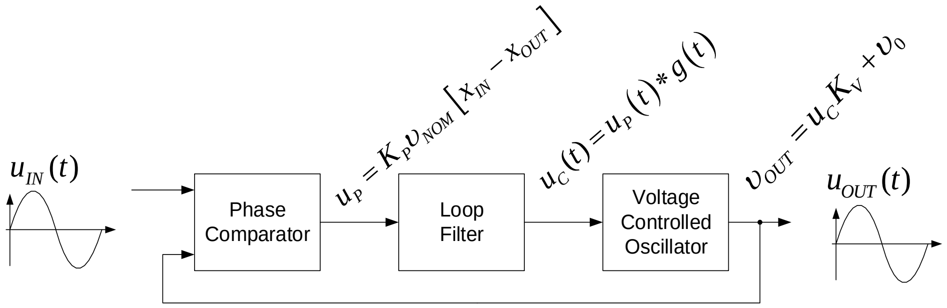
WHAT DOES A PLL DO?

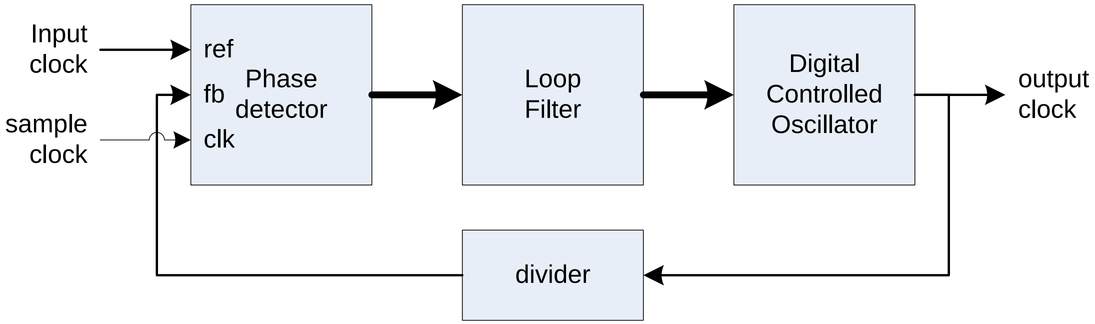
Generate a clock that is phase and frequency locked to an input clock.

- The output clock frequency can be of the same frequency, an integer multiple or a fraction of the input clock frequency.
- Input edge to output edge phase alignment can be achieved.

Input clock frequency to output clock frequency ratio must be a positive integer or fractional number.

- 19.44 MHz to 66/64*255/237*78125/77760*622.08 MHz is possible.
- 10 MHz to π MHz is not possible (afaik).

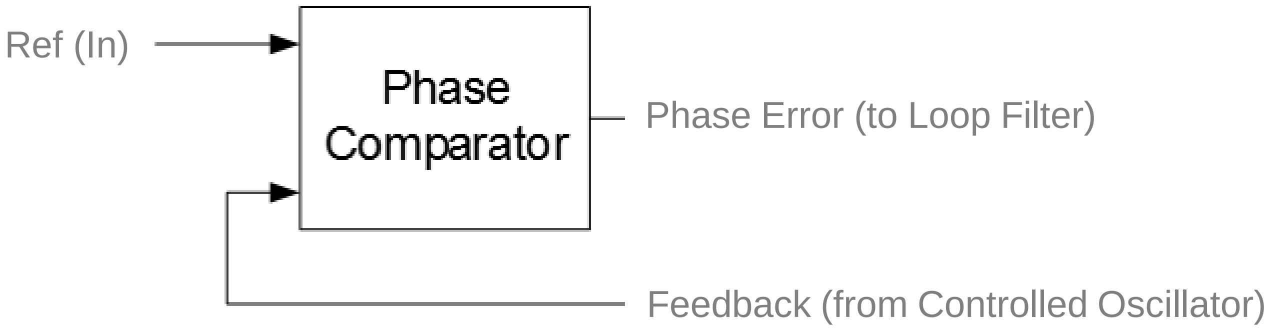

$u_{IN}(t)$
Phase Comparator
$u_P = K_P \upsilon_{NOM}\left[x_{IN} - x_{OUT}\right]$
Loop Filter
$u_C(t) = u_P(t) * g(t)$
Voltage Controlled Oscillator
$\upsilon_{OUT} = u_C K_V + \upsilon_0$
$u_{OUT}(t)$



$$u_{IN}(t) = A \cdot \sin\left\{2\pi\upsilon_{NOM}\left[t + x_{IN}(t)\right]\right\} = A \cdot \sin\left\{2\pi\upsilon_{IN}(t) + \varphi_{0,IN}\right\}$$

$$u_{OUT}(t) = A \cdot \sin\left\{2\pi\upsilon_{NOM}\left[t + x_{OUT}(t)\right]\right\} = A \cdot \sin\left\{2\pi\upsilon_{OUT}(t) + \varphi_{0,OUT}\right\}$$

PLL BUILDING BLOCKS: PHASE COMPARATOR


Ref (In)
Phase Comparator
Phase Error (to Loop Filter)
Feedback (from Controlled Oscillator)

The Phase Comparator establishes the “error” between the reference input and the clock output; using a feedback

Most Digital PLLs (DPLLs) use a Time to Digital Converter (TDC) for the Phase Frequency Detector (PFD) to measure the phase of the two clocks and produce a digital word representing the error

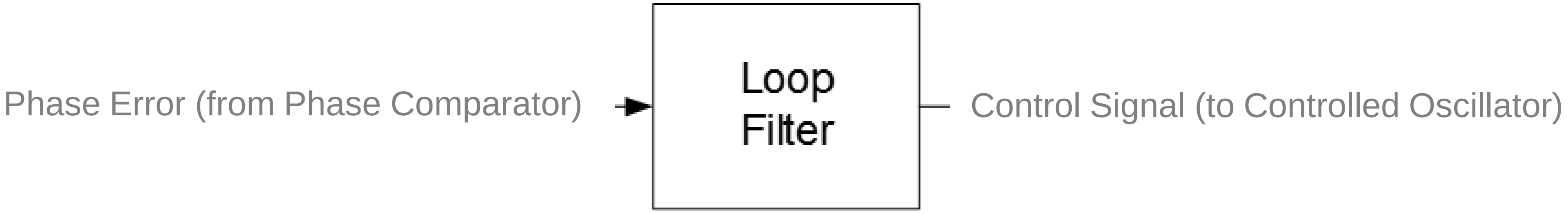
- TDC can be looked at as a timestamper, resolution determined by the sampling clock

The TDC timestamps the reference and feedback edges, and the PFD mathematically tracks the phase offset between the selected reference and feedback clocks

The measured phase difference can go well beyond 1 period, or Unit Interval (UI), of the reference and feedback clocks; thus, the phase comparator must be able to measure over a large range of multiple input/feedback clock periods

- Various telecom standards define a jitter & wander tolerance requirement - the widest is defined is 18µsp-p
- For example, if the input clock has a nominal period of 8ns (125 MHz), then the jitter tolerance requirement equates to ± 1125 UI

PLL BUILDING BLOCKS: LOOP FILTER



Phase Error (from Phase Comparator)

Loop Filter

Control Signal (to Controlled Oscillator)

The phase error is processed by the loop filter (LF)

- LF is a combination of proportional and integral (PI) control, which generates a control signal for controlling the oscillator
- The integrator is an additional pole, therefore 2nd order (may be referenced as “Type 2” PLL)

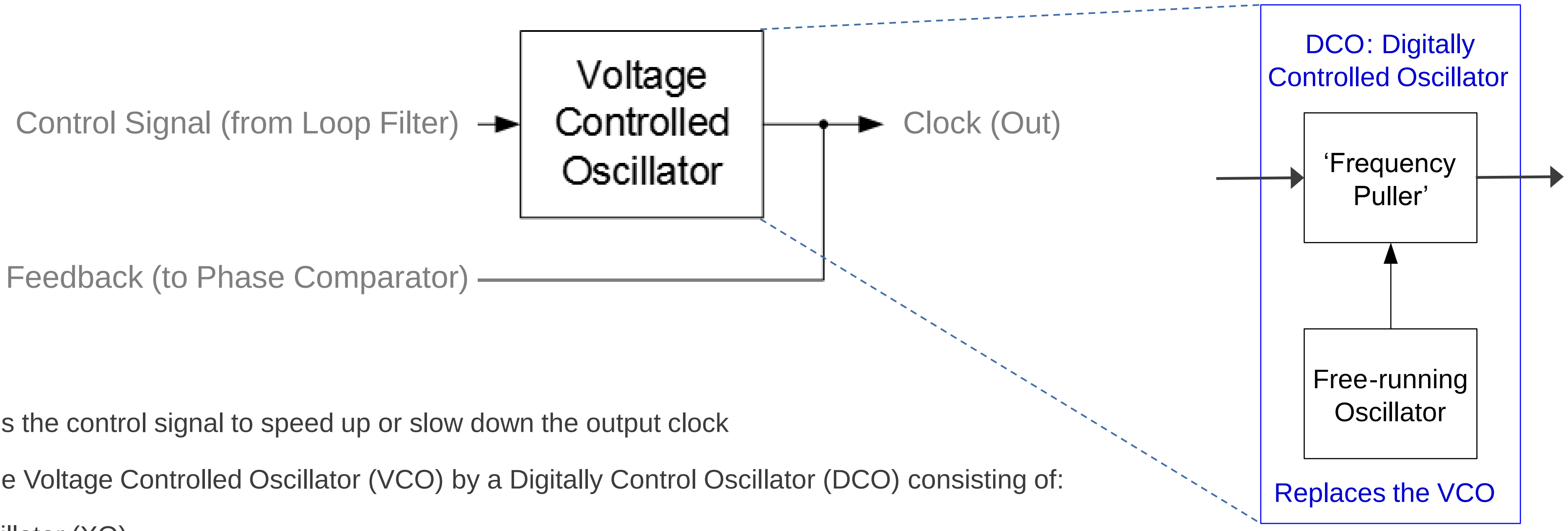
The LF determines the bandwidth (BW) of the PLL (i.e. cut-off frequency)

- Other functionality, such as phase slope limiting (PSL), locking range, and holdover functionality may be done as well

The step response has an overshoot and the frequency domain transfer function has peaking.

- Phase corrections mainly done through proportional path, along with any PSL
- Frequency offset, or drift corrections, is done through the integrator path, including damping (i.e. gain peaking control)

PLL BUILDING BLOCKS: CONTROLLED OSCILLATOR

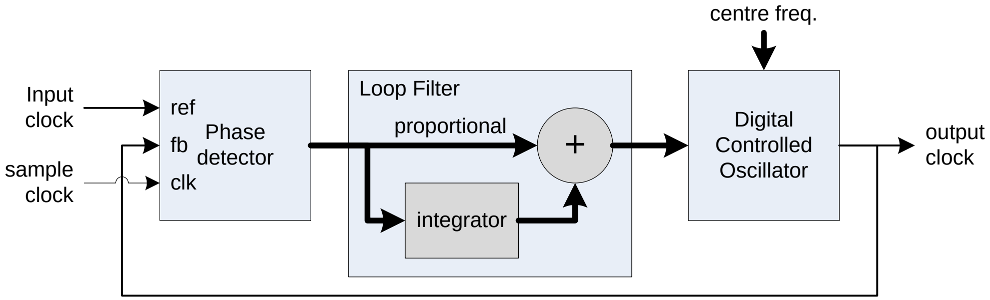

Control Signal (from Loop Filter)
Voltage Controlled Oscillator
Clock (Out)
Feedback (to Phase Comparator)
DCO: Digitally Controlled Oscillator
'Frequency Puller'
Free-running Oscillator
Replaces the VCO

The controlled oscillator uses the control signal to speed up or slow down the output clock

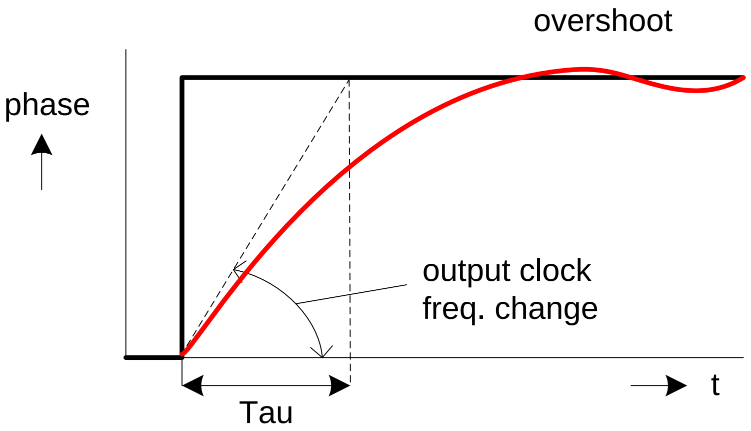
Most Digital PLLs replace the Voltage Controlled Oscillator (VCO) by a Digitally Control Oscillator (DCO) consisting of:

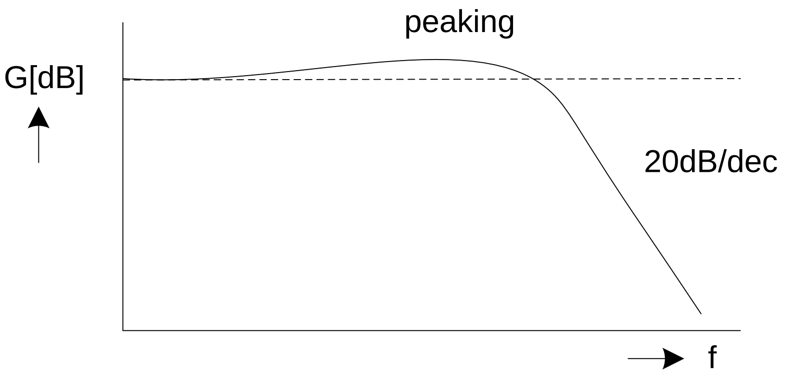
- a free-running crystal oscillator (XO)
- A digital synthesizer which pulls the frequency up or down using a Control Signal from loop filter (a digital word representing a fractional frequency offset (FFO))

TYPE 2 DPLL

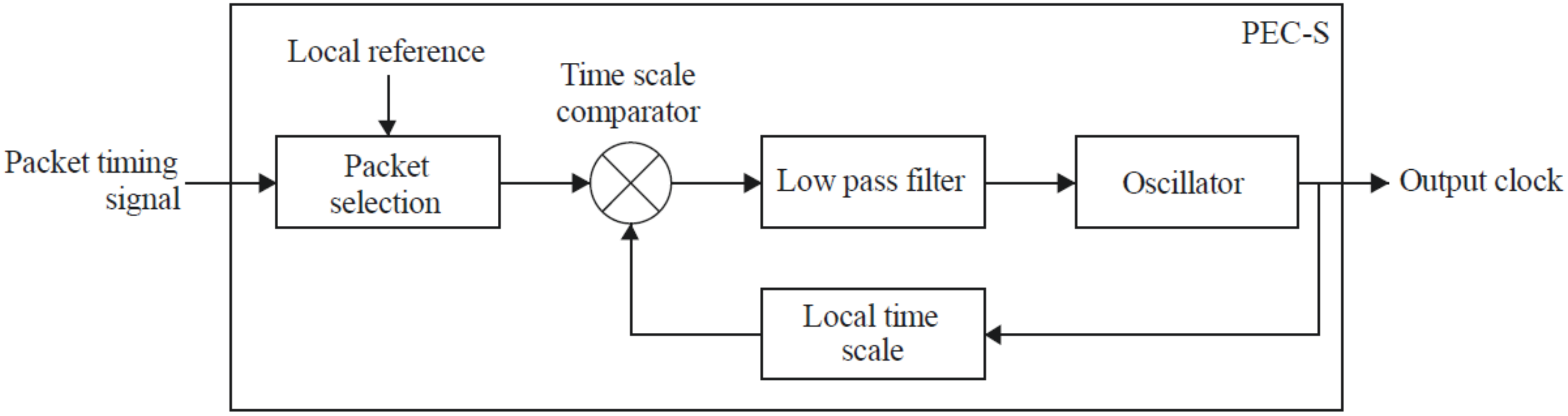


- A loop filter with an additional integrator is inserted.
- The integrator is an additional pole, therefore 2nd order.
- The DCO pole and the transfer function constants of the PFD and DCO act as a low pass filter.
- The jitter transfer function has only 20dB/dec roll-off.
- Type 2 DPLL. A non-zero delta freq. no longer requires a non-zero phase offset. So at a non-zero frequency offset the DPLL will lock with zero phase offset.
- The step response has an overshoot and the frequency domain transfer function has peaking.





PLL: USE WITH PACKET CLOCKS



G.8263-Y.1363(12)_FA.1

A timing protocol, such as IEEE 1588, can be used as the phase (or time) comparator

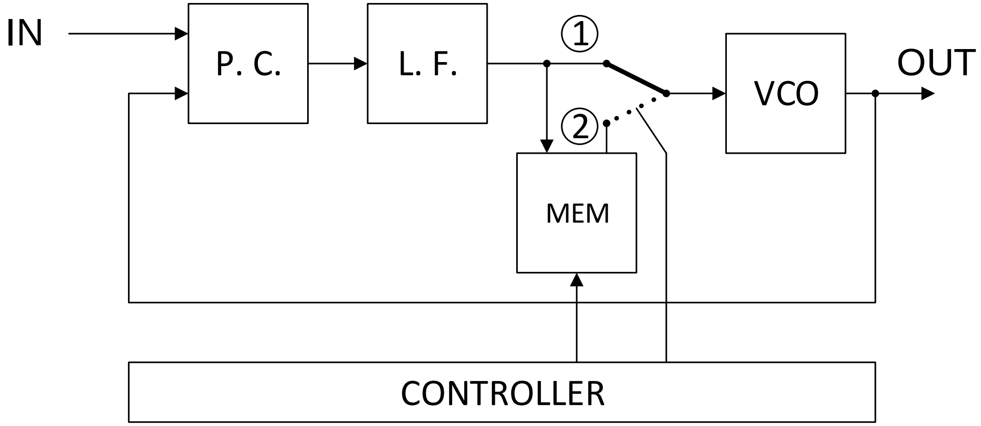
- Still follows same model as PLL, but may introduce a packet selection block
- Without packet selection, the packet delay variation (PDV) will have a significant impact to the loop filter

The LF will typically be designed to support much lower update intervals of the phase error

- This is due to packet dropping, or to attenuate the impact of the PDV
- Typically requires a more stable local clock source (oscillator, or maybe physical layer assistance)

PLL MODES: LOCKED, HOLDOVER AND FREERUN

- Freerun mode; the DPLL does not track any input clock. The output clock is at the centre frequency, offset is zero.
- i.e. switch is open.
- Normal / Lock mode; the DPLL tracks the input clock. The output clock is phase & frequency locked to the input clock.
  - i.e. switch is in position 1.
- Holdover mode; Typically used when the input clock fails. The PLL no longer tracks its input clock but uses the last valid frequency offset from memory (MEM). The proportional path is reset to zero and the integrator frozen at its last value. The phase detector is reset to flush out its phase history.
  - i.e. switch is in position 2.
  - clock becomes an autonomous synchronization source



In freerun mode or after entry into holdover mode, frequency is subject to ageing drift and to the influence of temperature.

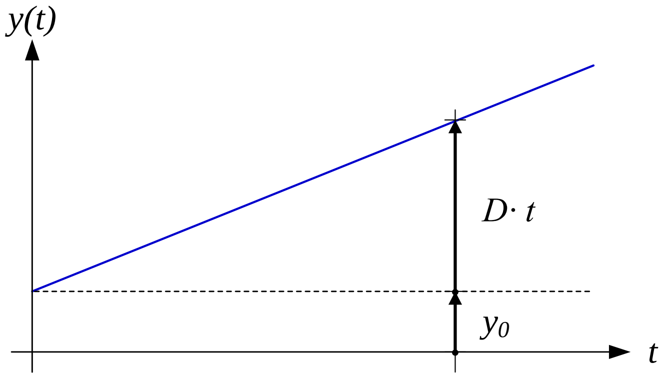
PLL MODES: HOLDOVER @ CONSTANT TEMPERATURE

Fractional frequency:

$$y(t) = y_0 + D \cdot t$$

where $y_0 =$ initial frequency offset

$D =$ frequency drift rate (constant)



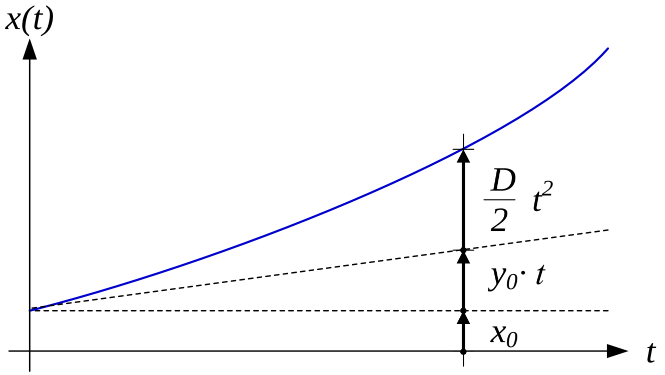
Time error:

$$x(t) = x_0 + y_0 \cdot t + \frac{D}{2} \cdot t^2$$

where $x_0 =$ initial phase offset

$y_0 =$ initial frequency offset

$D =$ frequency drift rate (constant)



LOOKING AT NOISE

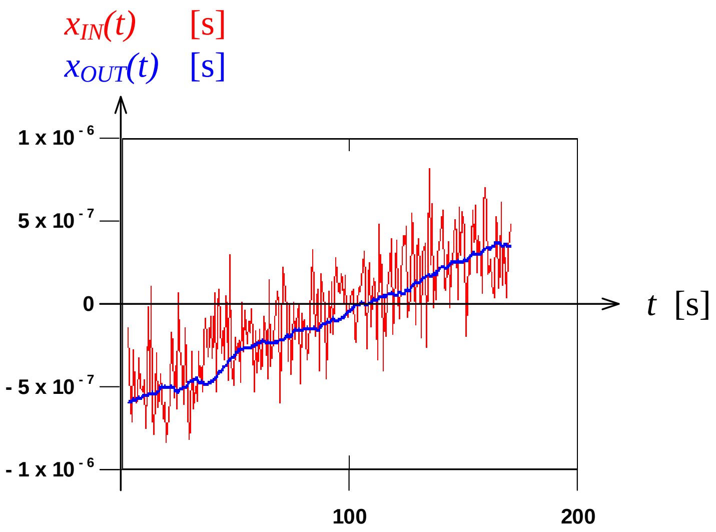
PLL: JITTER & WANDER FILTERING

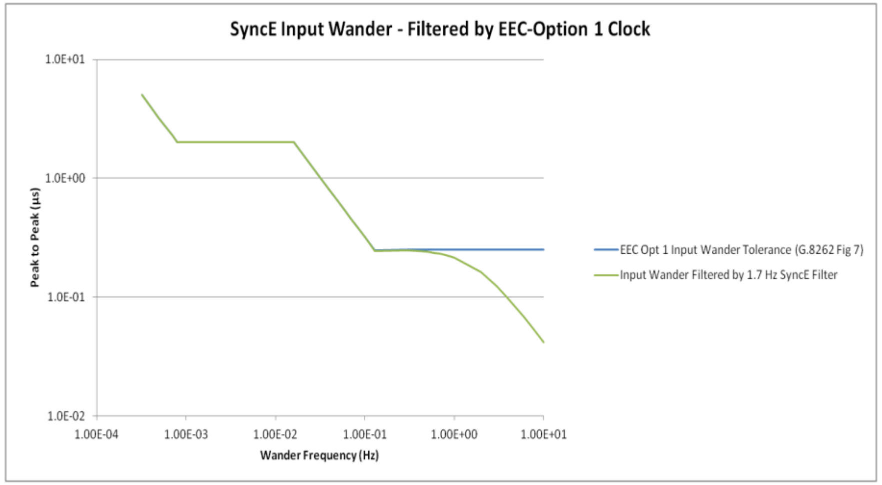
What is jitter/wander?

- Jitter, wander, phase noise is a variation of the clock's frequency/period/phase
- Essentially a phase modulation (due to noise or other disturbances) of the carrier clock when compared to an ideal reference
  - Jitter = short-term variations
  - Wander = long-term variations
- ITU-T G.810 defines noise frequencies <10Hz as wander and frequencies ≥10Hz as jitter
  - In Telecom, the period of the clock is called Unit Interval (UI)

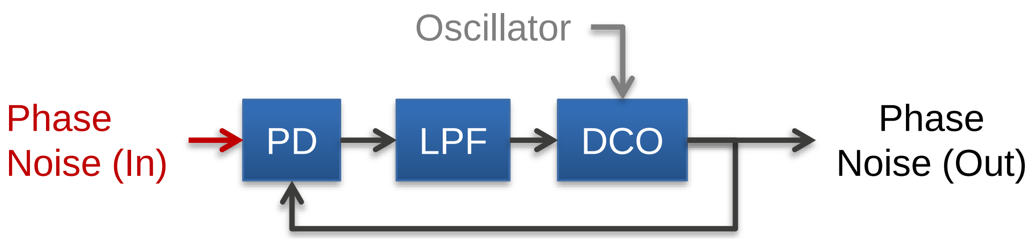
The function of a PLL is to attenuate jitter and transfer wander

- In other words, tolerate noise at the input without losing lock to the reference





PLL: RESPONSE TO INJECTED NOISE

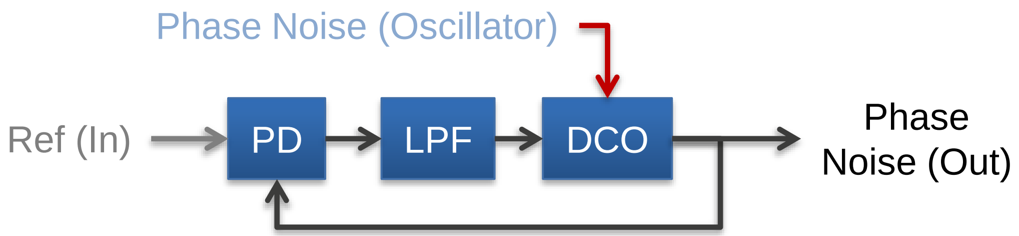


$$x_{OUT}(t) = x_{IN}(t) * h_{IN}(t)$$

$$X_{OUT}(s) = X_{IN}(s) \cdot H_{IN}(s)$$

where $h_{IN}(t) =$ impulse response

$H_{IN}(s) =$ transfer function $= \text{Laplace}\{h_{IN}(t)\}$



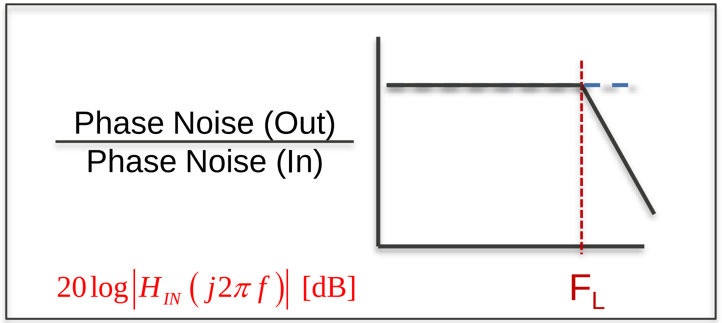
$$x_{OUT}(t) = x_{OSC}(t) * h_{OSC}(t)$$

$$X_{OUT}(s) = X_{OSC}(s) \cdot H_{OSC}(s)$$

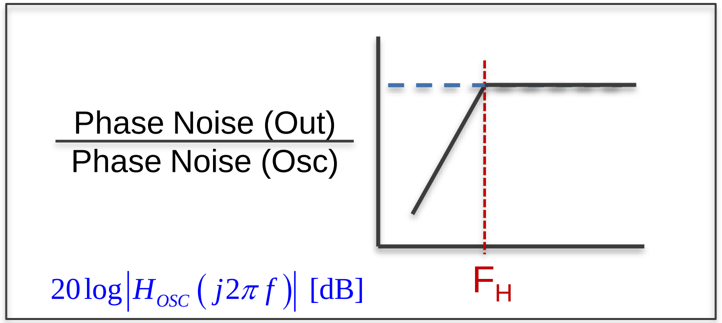
where $h_{OSC}(t) =$ impulse response

$H_{OSC}(s) =$ transfer function $= \text{Laplace}\{h_{OSC}(t)\}$

PLL is a low-pass filter for input noise

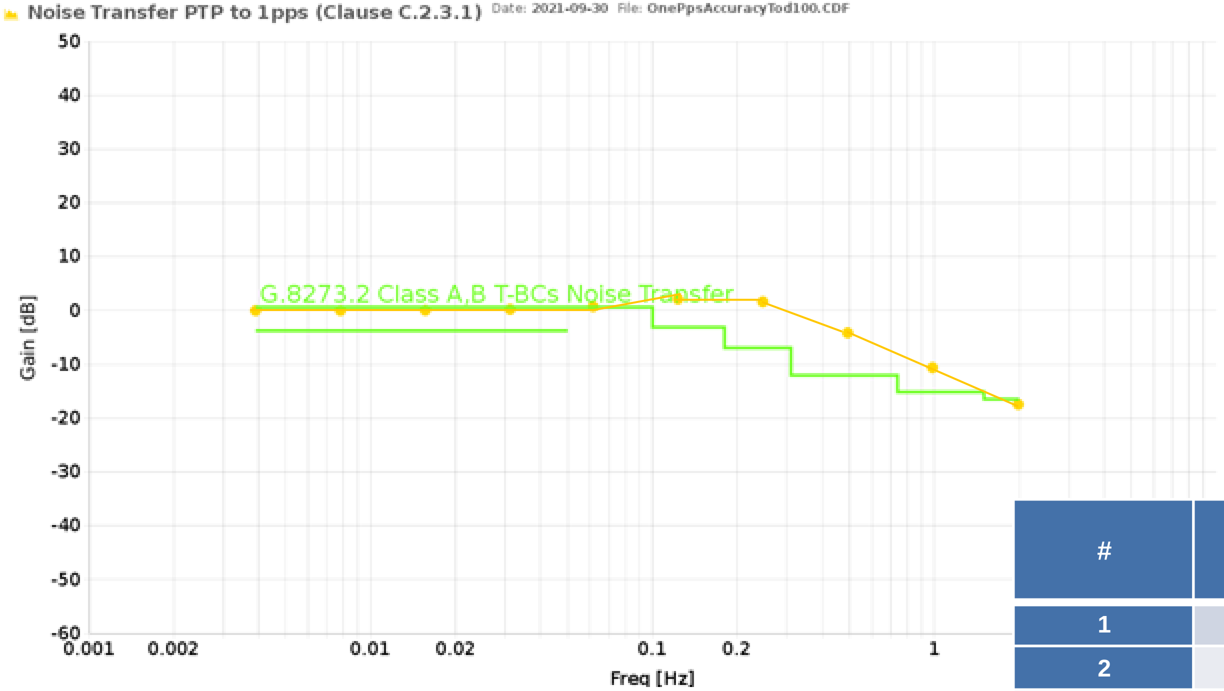


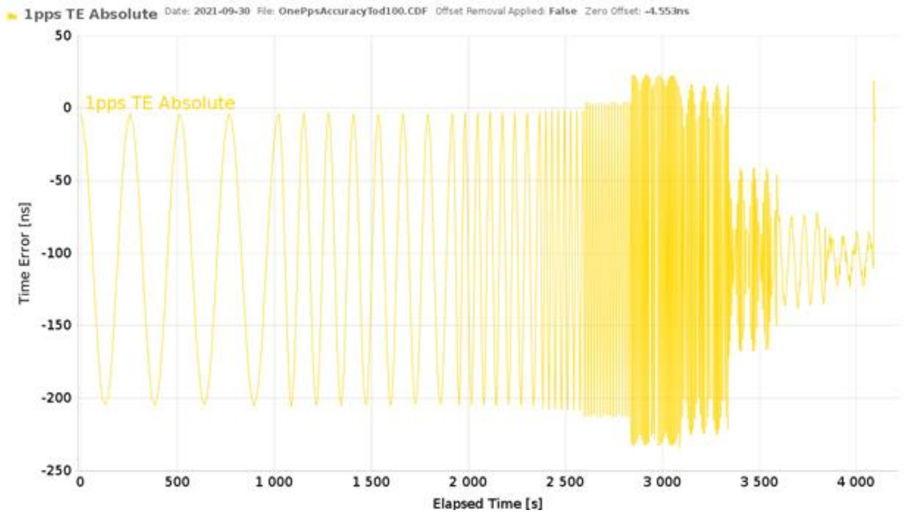
PLL is a high-pass filter for oscillator noise



PTP4L SW FILTER TEST RESULTS FOR G.8273.2

Simple SW based Low Pass Filtering (BW = 0.4-0.5Hz), no PDV





#	Freq (Hz)	Ampl (ns)	Duration (s)	Pk-Pk Output (ns)	Gain (dB)	Upper Pk-Pk Limit (ns)	Lower Pk-Pk Limit (ns)
1	0.00391	200	1024	199.77	-0.01	215	130
2	0.00781	200	896	200.11	0.00	215	130
3	0.01563	200	448	201.19	0.05	215	130
4	0.03125	200	224	205.06	0.22	215	130
5	0.06156	200	243.65	217.68	0.74	215	-
6	0.12313	200	251.78	255.64	2.13	140	-
7	0.24625	200	251.78	240.95	1.62	90	-
8	0.4925	200	249.75	124.11	-4.14	50	-
09	0.985	200	249.75	58.95	-10.61	35	-
10	1.985	200	249.87	26.83	-17.45	30	-

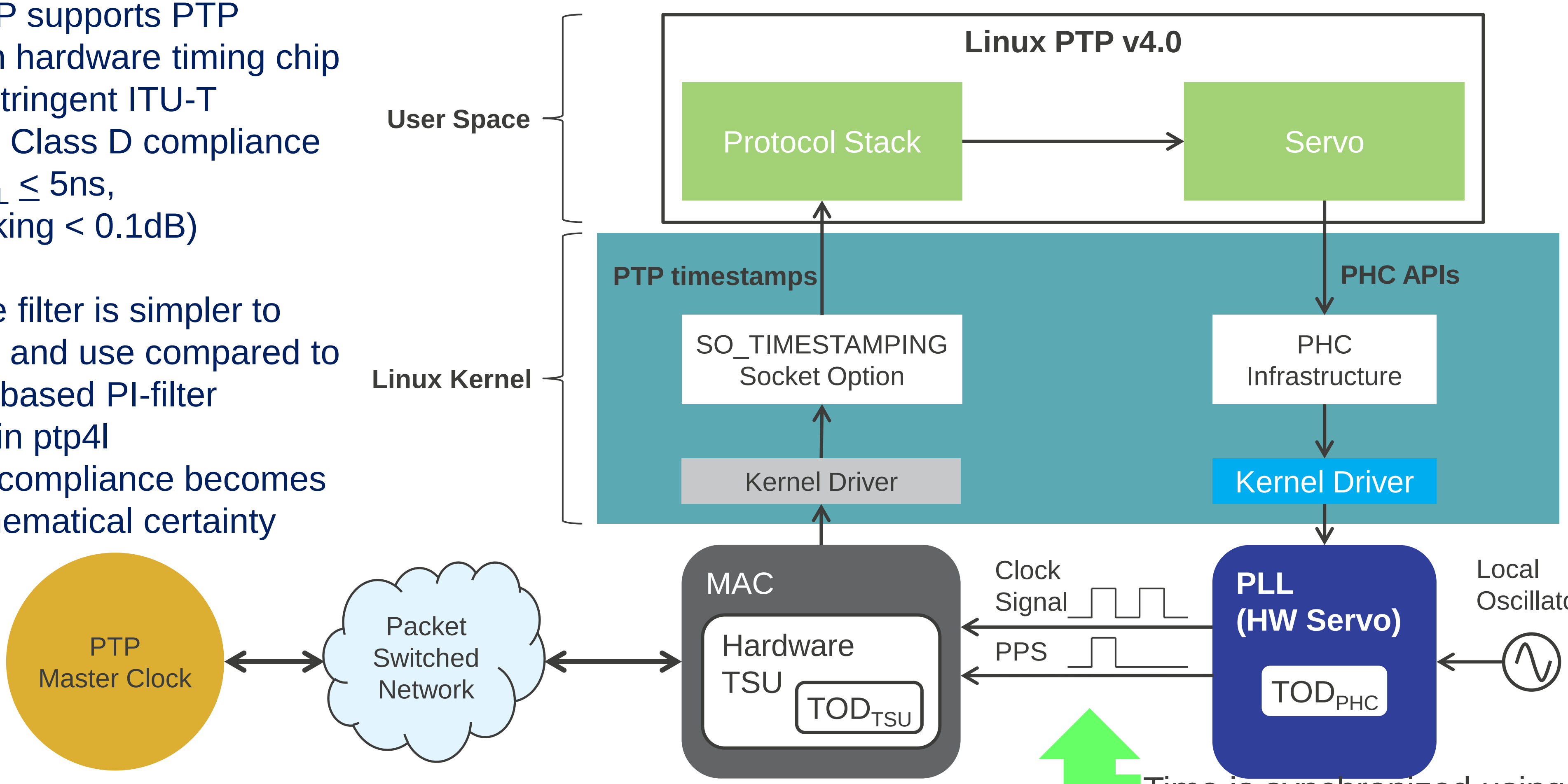
Even if ptp4l settings adjusted for cut-off frequency (i.e. to be <0.1Hz), it will still be difficult to pass gain peaking (>2db!)

DPLLS BECOMING COMPATIBLE WITH THE LINUX PTP HW CLOCK

Linux PTP supports PTP filtering in hardware timing chip to meet stringent ITU-T G.8273.2 Class D compliance ($\max|TE|_L \leq 5$ns, gain peaking < 0.1dB)

Hardware filter is simpler to configure and use compared to software-based PI-filter included in ptp4l

- Servo compliance becomes a mathematical certainty

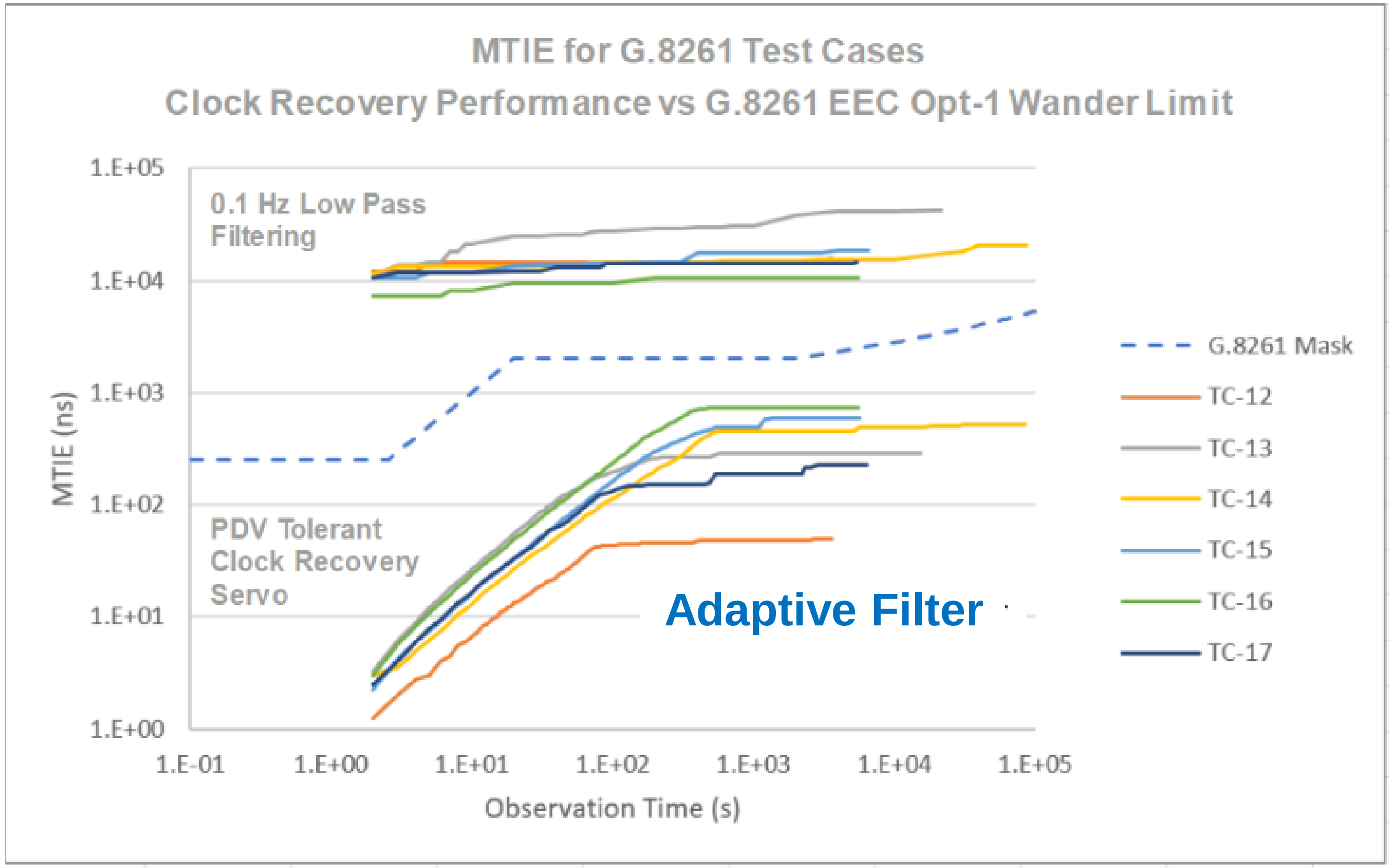


Same APIs

Drivers in Main Line Linux

Time is synchronized using the Linux PTP ts2phc application

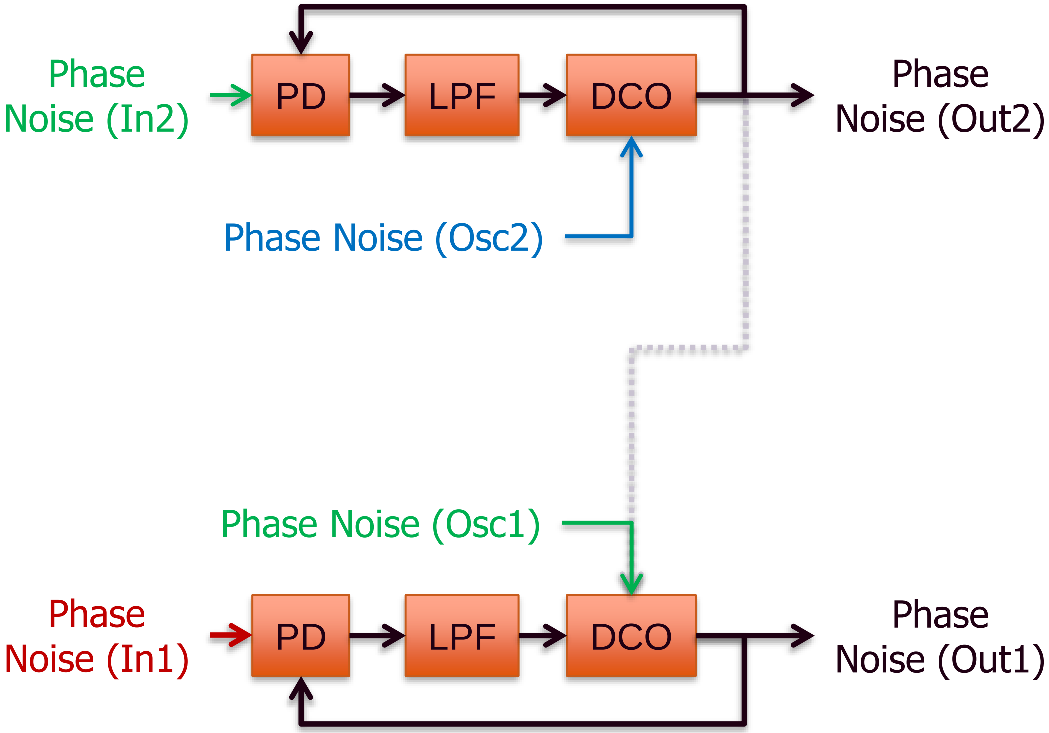
BENEFIT OF AN ADAPTIVE FILTER

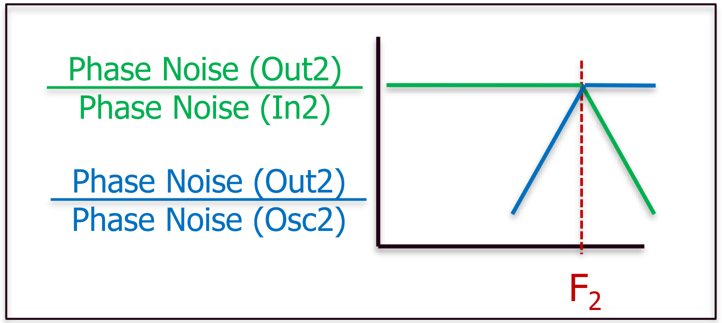


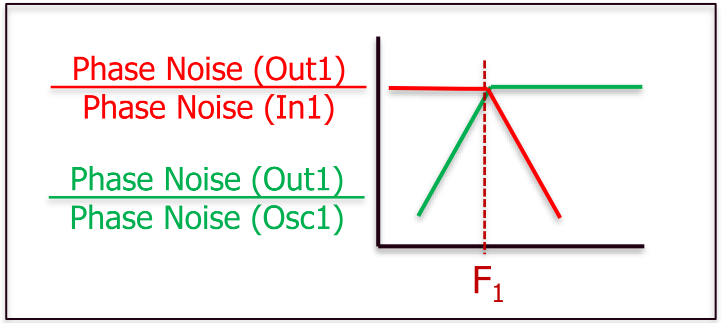
PHASE LOCKED LOOPS (PLL)

CLOCK COMBINING

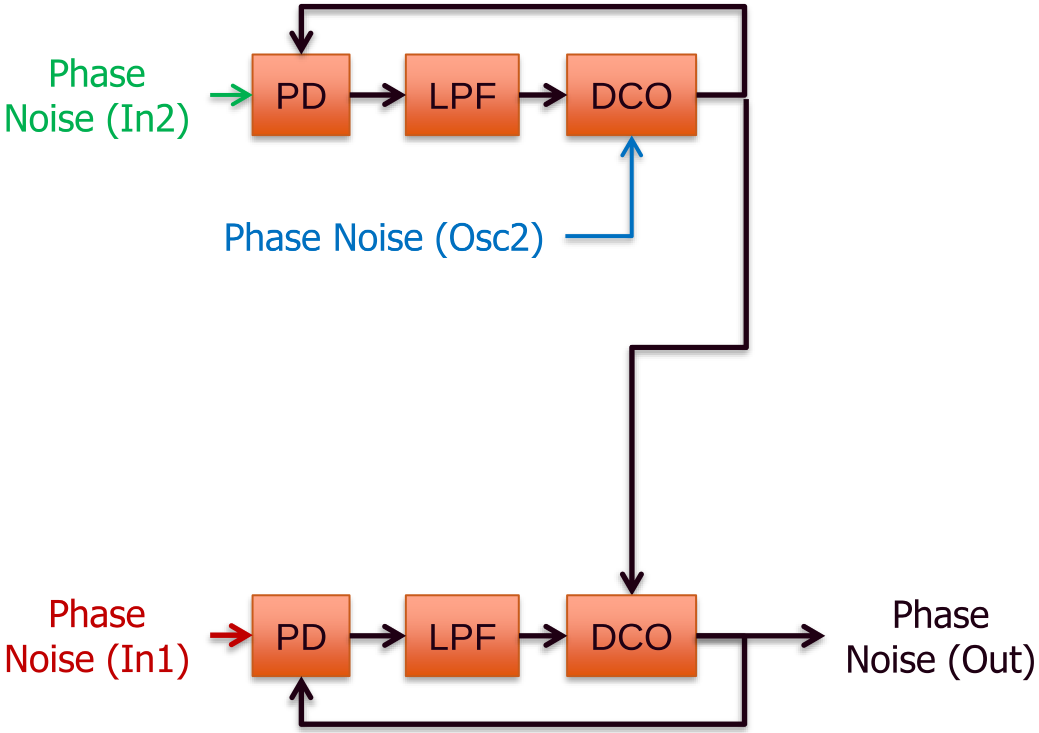
COMBINING TWO PLLS



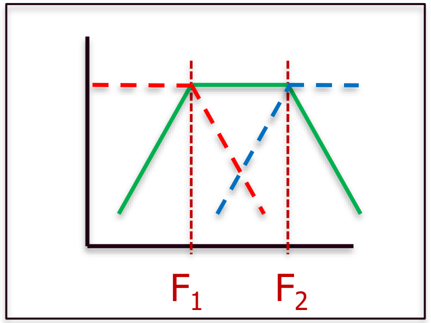




TWO PLLS: RESPONSE TO INJECTED NOISE

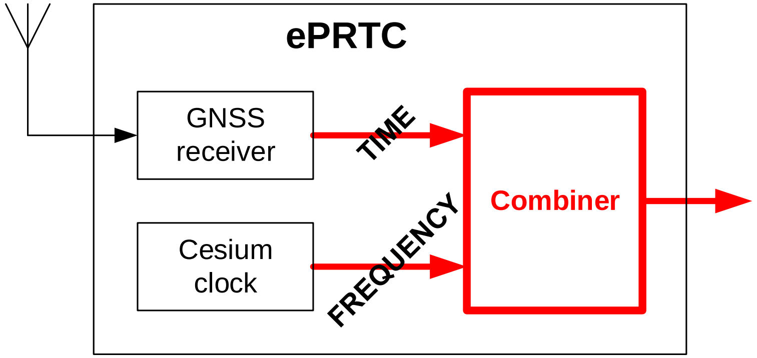


Band-pass filter for In2 Noise to Out

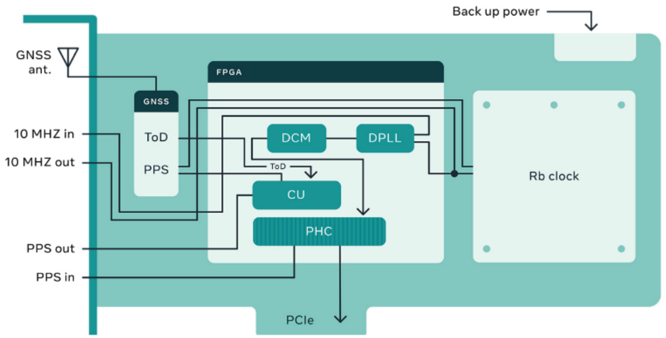


TWO PLLS: APPLICATIONS

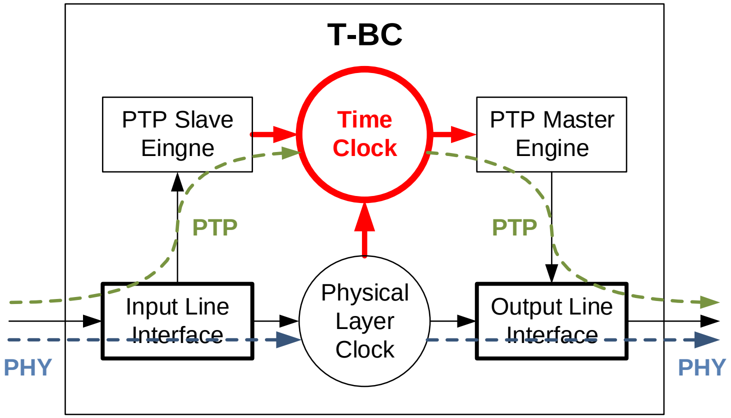
- **GNSS & Cesium clock** in enhanced Primary Reference Time Clocks (ePRTC)



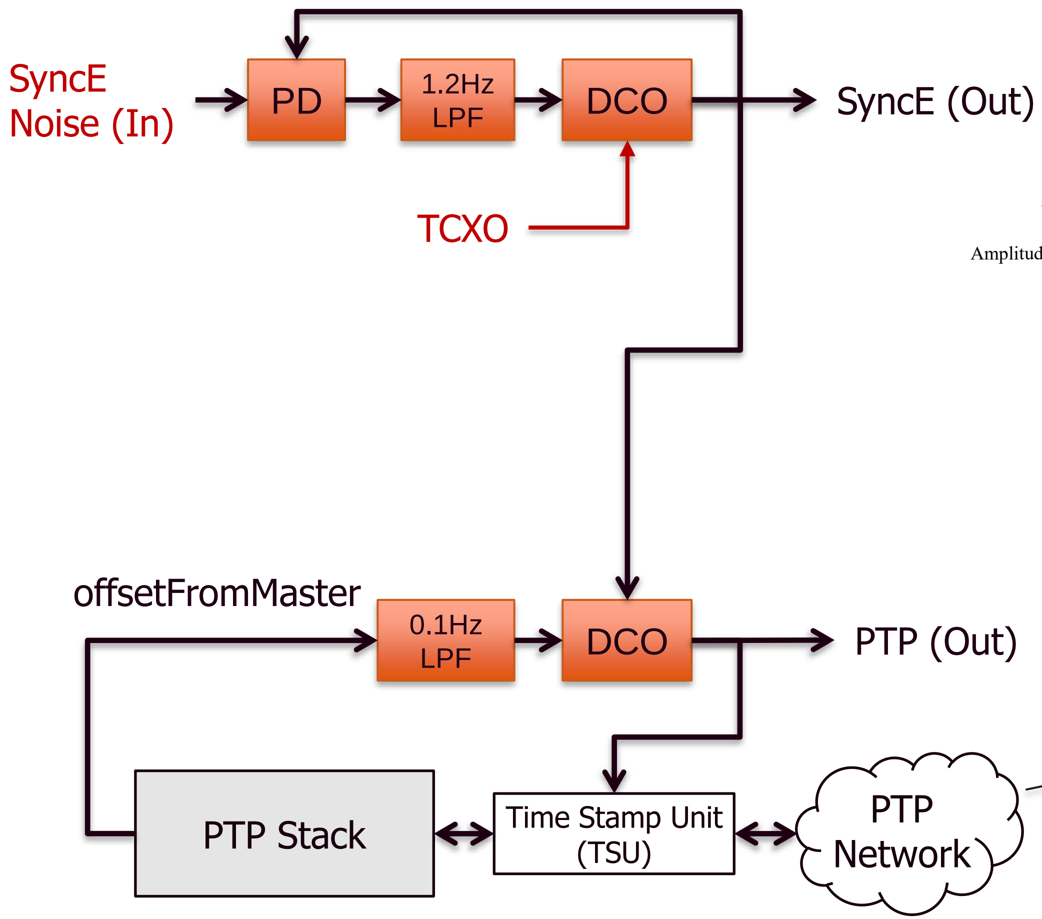
- **GNSS & Miniature Atomic Clock (MAC)** in OCP-TAP Time Card



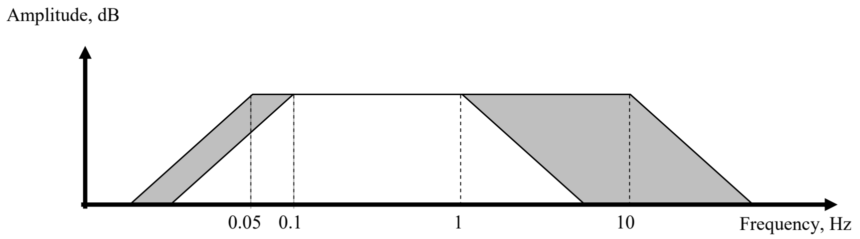
- **PTP & SyncE** in boundary clocks (T-BC) and slave clocks (T-TSC)



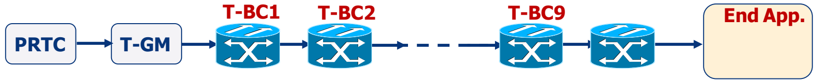
G.8273.2 T-BC: RESPONSE TO INJECTED SYNCE NOISE



A band-pass filter for SyncE Noise to PTP (Out)



Very little PDV



Renesas.com

Timing is the heartbeat of the system