

Time synchronization

IEC 61850 requires PTP (precision time protocol) synchronization between the different physical and virtual machines.

In the context of IEC 61850, it is used for communication between the machines (with SV, GOOSE or MMS protocol). In SEAPATH, this synchronization is also used between the different machines to form the cluster.

PTP is a Transmitter/Receiver architecture. It uses a `Grand master clock`, a physical device that will be the reference time. All other machines and devices synchronize themselves with it.

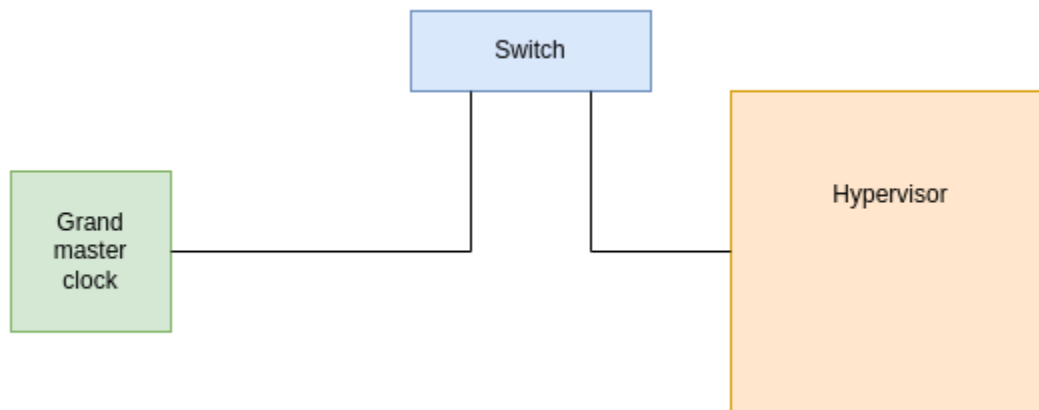
Connectivity

The grand master clock and the physical machines are connected with an Ethernet cable. In a very simple scenario, the cable can be plugged directly from the clock to the machine.

If a switch is required, the best is to use a PTP-compliant switch, which will assist clocks in measuring and adjusting for packet delay. These are called transparent clocks.

A classic switch can also be used for testing, but the timing will not be as good, and multiple switches cannot be used in cascade.

Below is a schema of the connectivity with a switch.

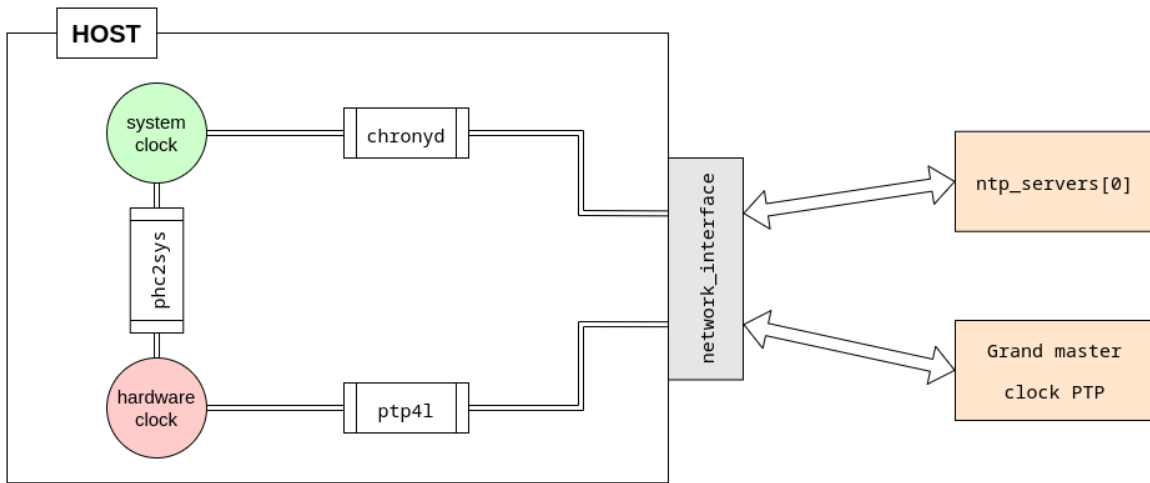


Hypervisor synchronization

The hosts synchronize the time with both NTP and PTP. This is done by the `timemaster` service during the network configuration section. (see [Network section](#)).

Three tools are used:

- `chrony` synchronize the system clock with NTP. The NTP server can be configured to be on the local network or on the internet.
- `ptp4l` synchronize the hardware clock (PHC) with PTP.
- `phc2sys` synchronize hardware and system clocks.



To see the status of the clocks and the delay with the references, use the command

```
chronyc sources
```

Logs of timemaster service are also useful to spot any errors:

```
systemctl status timemaster
```

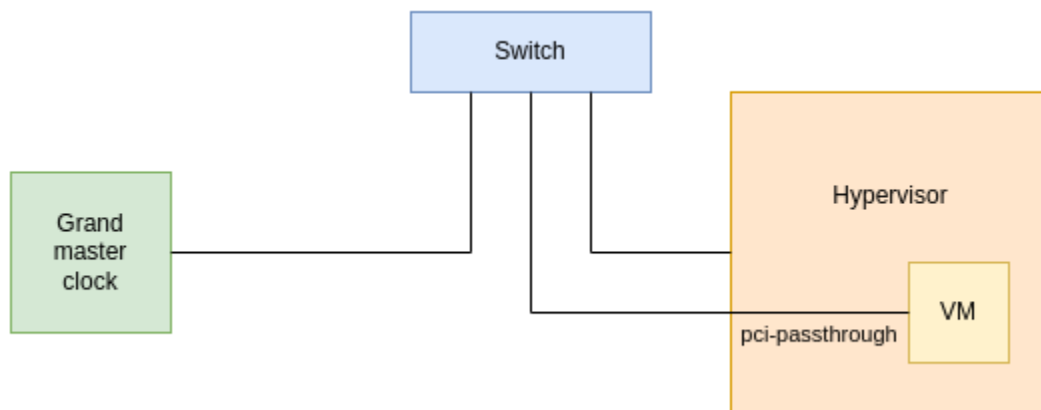
Virtual machine synchronization

The VM's clock can be synchronized in two ways.

PCI-passthrough

The network interface that receives PTP frames from the grand master clock can be configured in PCI-passthrough in order to be fully available inside the virtual machine. When doing that, a PTP clock will automatically be created inside the VM, it can be managed by timemaster. Refer to the hypervisor section.

Be aware that in this case, two interfaces must receive PTP frames: one to synchronize the host and one to synchronize the virtual machine.



PTP_KVM

Linux can make the PTP clock of the hypervisor available inside the virtual machine. This is done by the kernel module `ptp_kvm`. The clock is derived from the hypervisor's and is thus already synchronized.

To achieve this, inside the VM, you must

- Load the kernel module `ptp_kvm`, either automatically at boot or manually.
This will create a ptp device in `/dev/`. To determine which device is the one created by `ptp_kvm`, simply check that the file `/sys/class/ptp/ptp*/clock_name` does indeed say "KVM virtual PTP".
In this example, our `ptp_kvm` device is `ptp0`.
- Synchronize the system clock with the PTP clock.

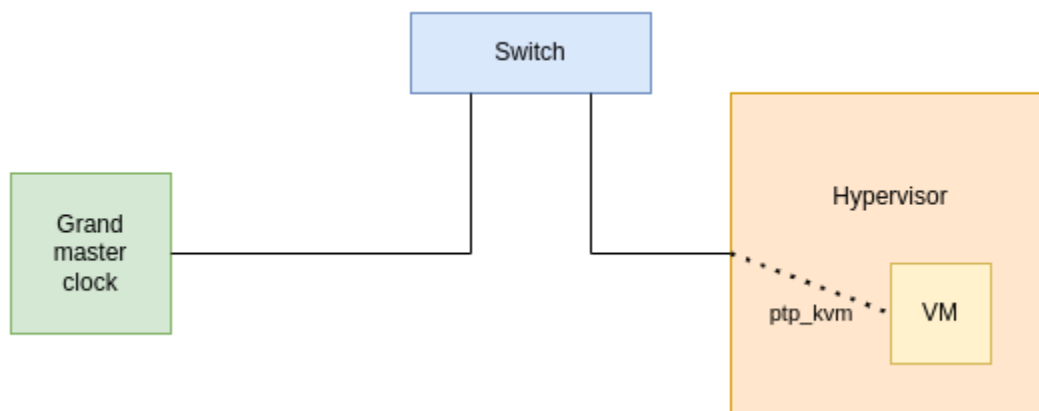
This can be done with the tool `phc2sys` from the `linuxptp` package :

```
phc2sys -s /dev/ptp0 -c CLOCK_REALTIME -O 0 -m -q
```

This command can be put in a `systemd` service to automatically synchronize the clocks at the start of the VM.

This can also be done inside the `chrony` configuration file with the command

```
refclock PHC /dev/ptp0 poll 3 dpoll -2 offset 0
```



Grand master simulation

Warning: This section is for test only. It is not intended to be used in a real environment.

To avoid the use of a physical grand master clock, a standard machine like a laptop can be used to simulate the PTP reference time. This is called « software PTP ».

To achieve that :

- Install `linuxptp` package on the laptop.
- Configure the PTP settings, in the file `/etc/linuxptp/ptp4l.conf`
- Starts the `ptp4l` service.

The configuration of PTP is fixed by IEC-61850 (see here on [Wikipedia](#)). Here is a proposed configuration of the file `/etc/linuxptp/ptp4l.conf` that fits the standard requirements.

```

[Global]
slaveOnly          0
gmCapable 1

domainNumber       0

# Announce interval: 1s
logAnnounceInterval 0

# Sync interval: 1 s
logSyncInterval    0

# Pdelay interval: 1 s
logMinPdelayReqInterval 0

# Announce receipt time-out: 3 s (fixed)
announceReceiptTimeout 3

priority1          128
priority2          128
# Default clock class : any specialised clock will be better (ie a GPS Grand Master Clock)
#clockClass        192
network_transport  L2
delay_mechanism     P2P
clockClass 248
clockAccuracy 0xFE
offsetScaledLogVariance 0xFFFF

```



The PTP clock created by ptp4l will not be synchronized with the current time. So even if your cluster machines are synchronized with each other, the date will be random.

To synchronize the PTP clock with the current time, type this command on the machine that simulate the grand master clock

```
phc_ctl <ptp_interface> set
```

Troubleshooting

rogue peer delay response

When a switch is used between PTP grand master clock and the hypervisors, if the timemaster service returns an error rogue peer delay response, this is probably because the switch doesn't support peer-to-peer PTP.

This error is particularly visible in a cluster : Connecting only one hypervisor is successful, but connecting another one fails.

To correct this behavior, you can

- Connect the hypervisor and the grand master clock directly (without a switch)
- Use a PTP compliant switch
- Change the delay_mechanism in the ptp4l configuration from P2P to E2E (end-to-end). Please be aware that this configuration is not the one required by IEC-61850, however, it can be useful for testing.