support

Timebeat / Cookbook / Starters and Apetisers

Articles in this section

How can I set up a simple PTP timing network

Last updated 2 months ago

This recipe will provide a step-by-step guide to setting up a basic timing network using Timebeat software. The goal of this process will be that at the end of which you can create your own simple PTP timing networks with the steps learned in this that are tailored and customised for your needs.

Problem:

You want to be able to set up a PTP grandmaster and have downstream servers talk PTP and synchronise to the UTC source.

Ingredients (the minimum required items)

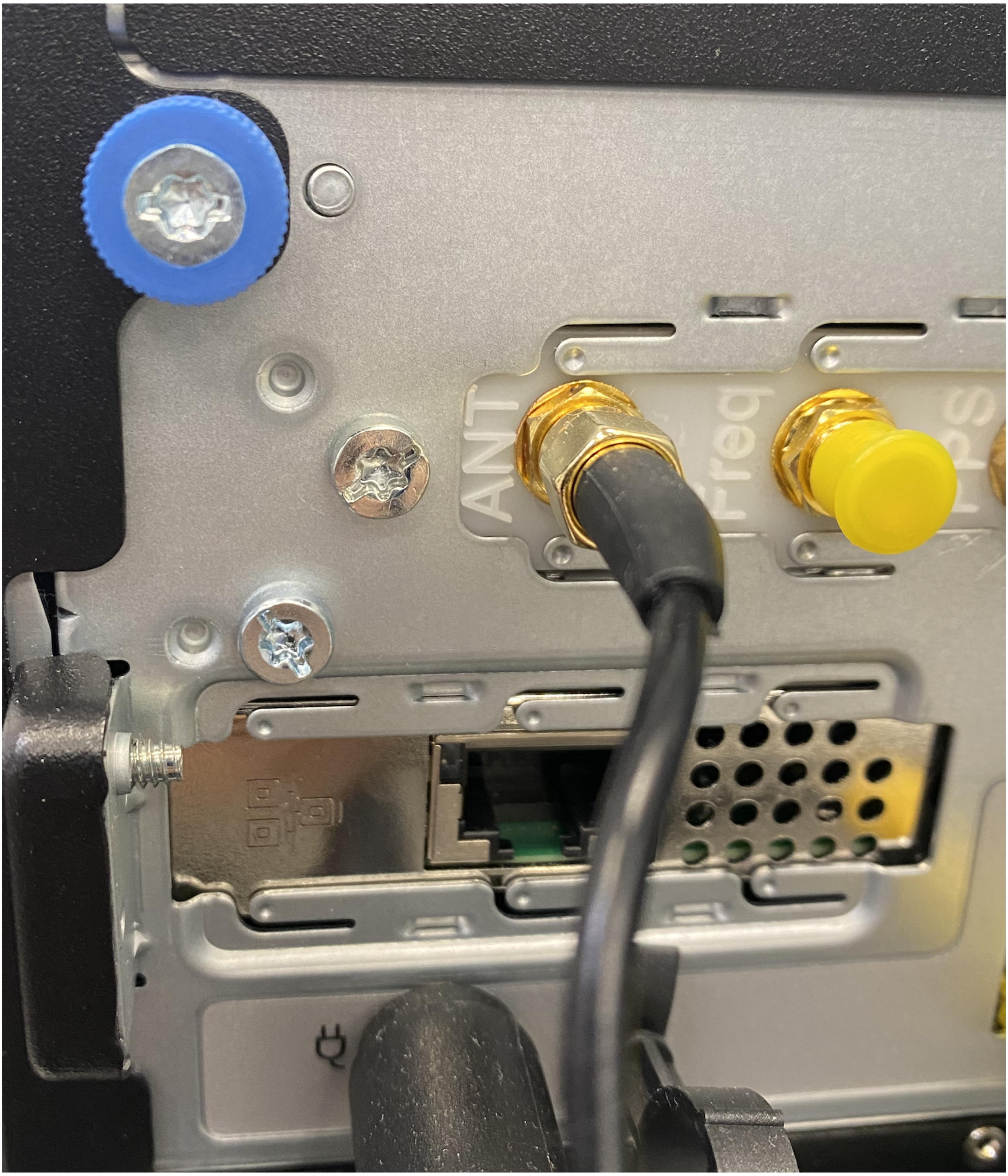
- A source of UTC
- A device operating as a PTP grandmaster
- A Local Area Network
- A device operating as a PTP slave
- Wireshark (tshark) - *optional*

Total Prep time: 15 mins, Total Cook time: 15 mins

In this recipe we will use a GPS antenna connected to a Timebeat Grandmaster Clock (G0kK-1) and Timebeat software

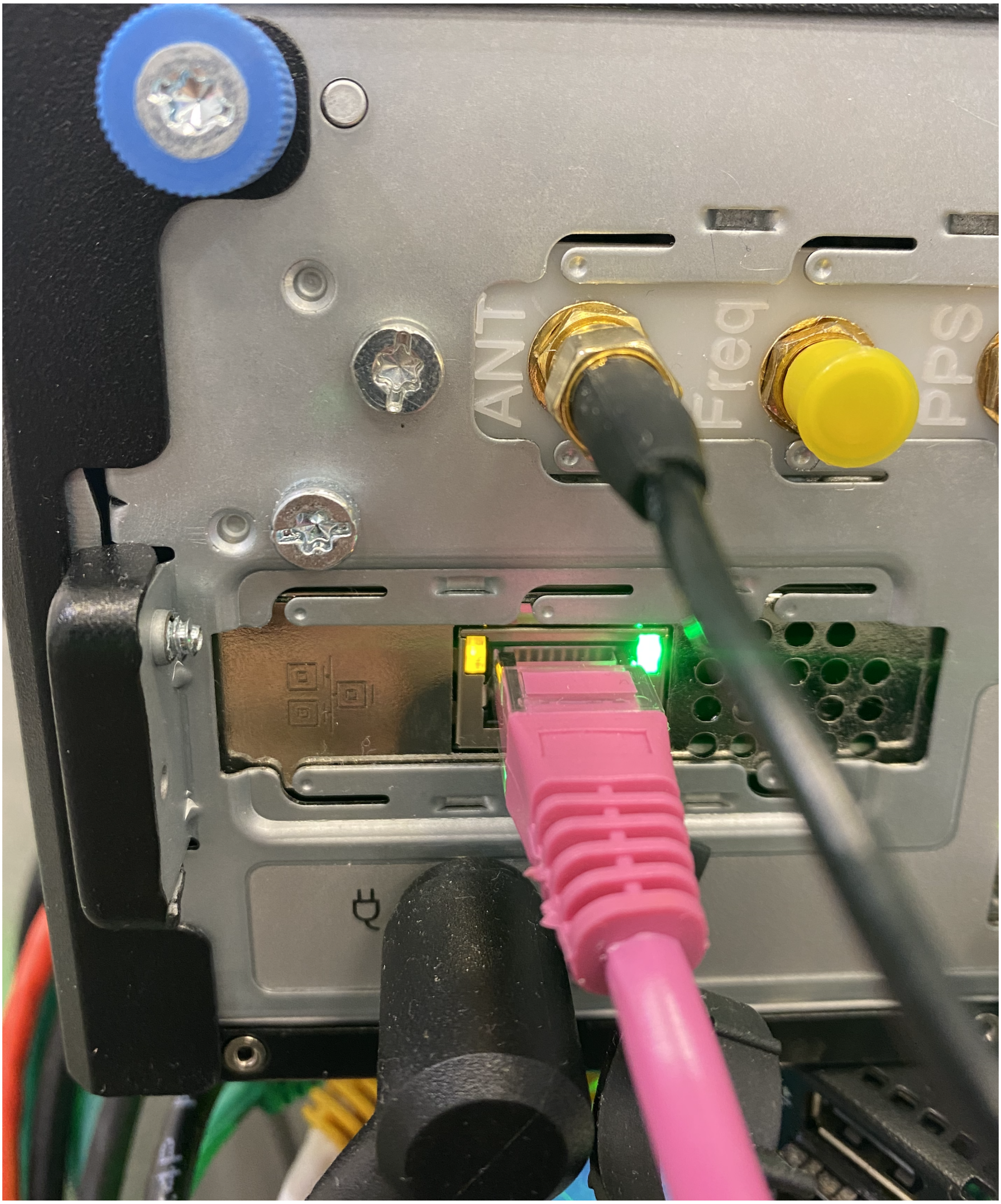
Step 1:

Connect the GNSS antenna to the Grandmaster



Step 2:

Connect ethernet cable to the appropriate network port (in this case we are connecting to ens1).



Step 3:

SSH or log into your grandmaster to configure the PTP feed. For timebeat software and grandmasters we SSH to the console and then access the config file:

LINUX:

```
root@gm01 ~]# vi /etc/timebeat/timebeat.yml
```

WINDOWS:

Open up the timebeat.yml file using notepad or similar from your installation directory (typically recommended program files --> Timebeat....)

Step 4:

(*All amended fields are highlighted*)

We will want to set up our PTP domain. For now, we will stick to the default of 0.

We then need to choose Unicast PTP distribution or Multicast. For this recipe, we will use multicast.

So we uncomment (remove the # from the start of the line) line 32

We will want to ensure that this device operates solely as a grandmaster

So we uncomment line 33

Lastly we will want to make sure PTP is delivered from the correct interface.
As we already cabled up interface ens1 lets just place that into the config.

For this, we uncomment line 44 and modify the interface name to read ens1

```
27 # PTP Config example
28      - protocol:                 ptp
29        domain:                   0
30        #serve_unicast:           true
```

```
31          #max_unicast_subscribers: 0
32          serve_multicast:         true
33          server_only:             true
34          announce_interval:        1
35          sync_interval:            0
36          delayrequest_interval:    0
37          #unicast_master_table:    ['1.2.3.4','2.3.4.5', '3.4.5.6
38          #delay_strategy:          e2e   # other options which ca
39          #hybrid_e2e:              false # Send delay requests as
40          #priority1:               128
41          #priority2:               128
42          #monitor_only:            false
43          #use_layer2:              false     # Use ptp over ethe
44          interface:               ens1
45          #profile:                 'G.8275.2' # other options whi
46          #logsource:               'Grandmaster Clock in NY4' # c
47          #asymmetry_compensation:  0 # In nanoseconds. Static com
48          #max_packets_per_second:  0 # If inbound packet rate exc
```

Step 6:

Now we will configure the source of UTC for the Grandmaster. In this case, we are using a Timebeat G0kK-1 grandmaster so we scroll down in the config file to the PPS section

So for this, we will uncomment every line from line 73 - 80

We want to make sure PPS is configured to the correct interface. In this case, it is ens1 so line 74 gets changed to ens1

In the G0kK-1 it is important to note PPS is delivered over Pin 1 and index 1 so we modify that on line 75 & 76

```
72    # 1 Pulse-per-second input config example. 1-PPS out can be co
73        - protocol:     pps
74          interface:    ens1
75          pin:          1
76          index:        1
```

```
          cable_delay:  0           # Cable delay in nanoseconds (c
          edge_mode:    "rising"    # PPS event trigger for older c
          monitor_only: false
          utc_offset:   37          # If TAI or similar UTC offset
```

It is important to note at this stage PPS only provides minor time, so we need to configure our secondary source to be an NMEA source to provide major time to the Grandmaster.

This step is super simple as all you need to do is uncomment lines 144-148 no changes are necessary.

```
141   # Other vendor specific GNSS receivers input config example
142      # Eight data bits, no parity bit, and one stop bit (8N1)
143      # (It is not recommended to configure a Mini-JLT source i
144      - protocol:     timecard-mini  # "mini-jlt" (Jackson Labs
145        device:       '/dev/serial0' # Serial device path
146        baud:         9600         # Serial device baud rate
147        offset:       0              # Static offset of RMC lin
148        monitor_only: false
```

Once complete just save the config and quit. (if using vi the below command will do the trick).

```
:wq!
```

Step 7:

Now we start the service using the standard commands

LINUX:

```
root@gm01 ~]# systemctl start timebeat
```

WINDOWS:

The easiest method in Windows is to open up the Task Manager, select Processes, find Timebeat in the list, right-click and select start.

Step 8 - *Optional*:

Let's check to see if PTP is leaving the correct interface. For this, we will just run a quick tshark command (provided by Wireshark, on Linux if you dont have this just run "dnf install wireshark-1:3.6.2-1.fc36.x86_64 - y" or alternatively run "dnf provides wireshark" and select the appropriate package from the list.

```
[root@gm01 ~]# tshark -i ens1 port 319 or port 320
```

note that -i represents interface and ens1 is the interface we wish to check

We should see output like the below:

```
[root@gm01 ~]# tshark -i ens1 port 319 or port 320
Capturing on 'ens1'
** (tshark:35277) 17:10:22.266280 [Main MESSAGE] -- Capture starte
** (tshark:35277) 17:10:22.266340 [Main MESSAGE] -- File: "/var/tm
1 0.000000000 10.101.103.31 → 224.0.1.129 PTPv2 106 Announce Messa
2 0.008392124 10.101.103.31 → 224.0.1.129 PTPv2 86 Sync Message
3 0.015557624 10.101.103.31 → 224.0.1.129 PTPv2 86 Sync Message
4 0.015636935 10.101.103.31 → 224.0.1.129 PTPv2 86 Follow_Up Messa
5 0.023994680 10.101.103.31 → 224.0.1.129 PTPv2 86 Sync Message
6 0.024078860 10.101.103.31 → 224.0.1.129 PTPv2 86 Follow_Up Messa
7 0.031399504 10.101.103.31 → 224.0.1.129 PTPv2 86 Sync Message
8 0.031464042 10.101.103.31 → 224.0.1.129 PTPv2 86 Follow_Up Messa
9 0.039642671 10.101.103.31 → 224.0.1.129 PTPv2 86 Sync Message
```

Step 9:

Now we will configure the slave device to receive PTP and synchronise towards.

For this we repeat Step 3 but using the details of the new device:

SSH or log into your receiving device to configure the PTP feed to be received. For timebeat software and grandmasters we SSH to the console and then access the config file:

LINUX:

```
root@localhost ~]# vi /etc/timebeat/timebeat.yml
```

WINDOWS:

Open up the timebeat.yml file using notepad or similar from your installation directory (typically recommended program files --> Timebeat....)

Step 10:

(*All amended fields are highlighted*)

We want to match the grandmasters PTP configuration in the receiving device.

We will want to set up our PTP domain. For now, we will stick to the default of 0 as this is also what our grandmaster has.

So all we need to amend here is the interface.

For this, we uncomment line 44 and modify the interface name to read ens1

```
27 # PTP Config example
28      - protocol:                 ptp
29        domain:                   0
30        #serve_unicast:           true
31        #max_unicast_subscribers: 0
```

```
        #serve_multicast:         true
        #server_only:             true
        announce_interval:        1
        sync_interval:            0
        delayrequest_interval:    0
        #unicast_master_table:    ['1.2.3.4','2.3.4.5', '3.4.5.6
        #delay_strategy:          e2e   # other options which ca
        #hybrid_e2e:              false # Send delay requests as
        #priority1:               128
        #priority2:               128
        #monitor_only:            false
        #use_layer2:              false      # Use ptp over ethe
        interface:               ens1
        #profile:                 'G.8275.2' # other options whi
        #logsource:               'Grandmaster Clock in NY4' # c
        #asymmetry_compensation:  0 # In nanoseconds. Static com
        #max_packets_per_second:  0 # If inbound packet rate exc
```

Once complete just save the config and quit. (if using vi the below command will do the trick).

```
:wq!
```

Step 11:

Now we start the service using the standard commands

LINUX:

```
root@gm01 ~]# systemctl start timebeat
```

WINDOWS:

The easiest method in Windows is to open up the Task Manager, select Processes, find Timebeat in the list, right-click and select start.

Step 12 - *Optional*:

Let's check to see if PTP is arriving on the correct interface. For this, we will just run a quick tshark command identical to Step 7

```
[root@fedora ~]# tshark -i ens1 port 319 or port 320
```

note that -i represents interface and ens1 is the interface we wish to check

We should see output like the below:

```
[root@fedora ~]# tshark -i ens1 port 319 or port 320
```

And that is a completed PTP timing network. Now all that is left is to investigate the performance.

Check out our next recipe on How to set up a monitoring solution and dashboard environment.

Was this article helpful?

Yes No

0 out of 0 found this helpful

Have more questions? Submit a request