syn1588 ®

Version 1.3 - January 2022

Abstract

After running the first tests using the syn1588® PTP Stack as described in the syn1588® PCIe NIC Quick Start Guide one starts setting up a real-life PTP system in a more complex scenario. This application note describes how to analyze the PTP port status using very simple means to identify frequent configuration issues like IP address mis-assignments, network issues or firewall configurations blocking the traffic.

Scenario

After having completed the first simple tests using the syn1588® PTP Stack most likely together with the syn1588® PCIe NIC following the Application Note “syn1588® PCIe NIC Quick Start Guide” (AN004, Version 1.6 - May 2019). Everything runs smoothly so far. But now you connect to your real-life network using your PTP Grandmaster.

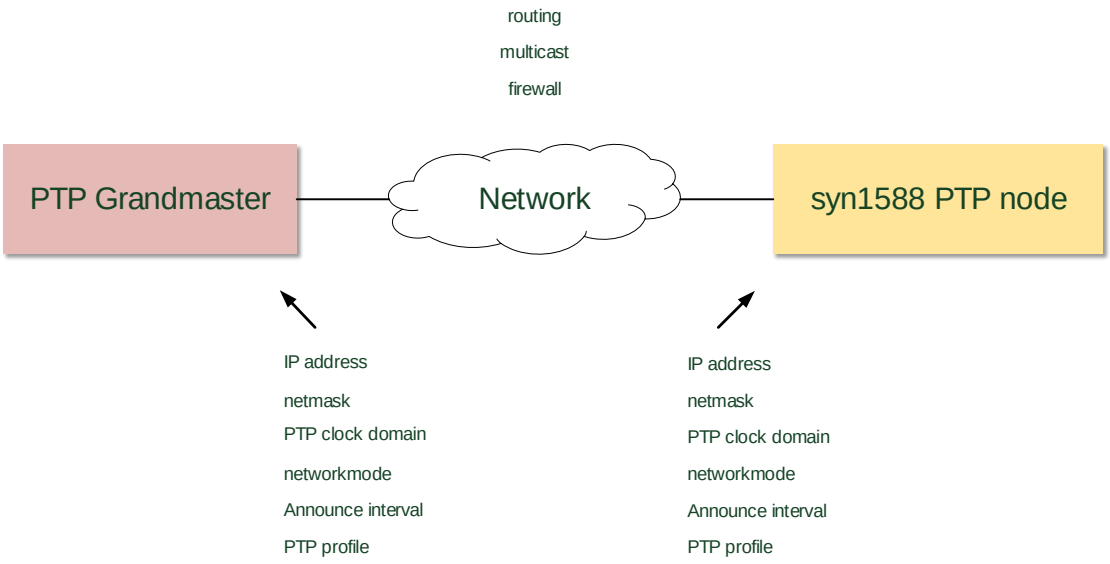

figure 1 Application scenario

Frequently you do not have direct access to the PTP Grandmaster in the network as well as the network infrastructure (switches etc.) since they are managed by your IT department. This application note provides you simple guidelines on how to identify connection or configuration issues using very simple means.

Your Tools

There are two very simple tools one can use to analyze the proper PTP communication on your node.

syn1588® PTP Stack Log Output

This “tool” comes free of charge with the syn1588® PTP Stack. The syn1588® PTP Stack provides a detailed log output with selectable verbosity level “-v <verbosity level>”. While for normal operation a log level of 1 or 2 is sufficient one is interested in all details (log level 4) during debugging or setup procedure.

You can easily figure out network or PTP configuration issues by analyzing the output. The following listing shows such a typical syn1588® PTP Stack log output.

```
oregano@KAEFER:/opt/oregano/bin$ sudo ./ptp -i enp2s0 -C S -v info -r 0
2022-01-05 15:48:48.845172 [INFO    ] [             ]  syn1588(R) PTP Stack -
IEEE1588-2008 Engine
2022-01-05 15:48:48.845206 [INFO    ] [             ]  Build: 2022-01-
05T12:43:16 - v1.14-9-g7cbf9e73
2022-01-05 15:48:48.845212 [INFO    ] [             ]  (c) Oregano Systems
- Design & Consulting GesmbH 2005-2022
2022-01-05 15:48:48.845217 [INFO    ] [             ]  Confidential unpublished data
- All rights reserved
2022-01-05 15:48:48.845224 [INFO    ] [             ]  syn1588(R) PTP Stack started:
2022-01-05 14:48:48.845222 (UTC)
2022-01-05 15:48:48.845317 [INFO    ] [p1           ]  Port 1: adding config "i" =
"enp2s0"
2022-01-05 15:48:48.845327 [INFO    ] [p1           ]  Port 1: adding config "C" =
"S"
2022-01-05 15:48:48.845333 [INFO    ] [p1           ]  Port 1: adding config "v" =
"info"
2022-01-05 15:48:48.845338 [INFO    ] [p1           ]  Port 1: adding config "r" =
"0"
2022-01-05 15:48:48.845344 [INFO    ] [p1           ]  PTP version 2.0
2022-01-05 15:48:48.845348 [INFO    ] [             ]  Command line: ./ptp -i enp2s0
-C S -v info -r 0
2022-01-05 15:48:48.845358 [WARNING ] [             ]  Failed to set high priority
for the PTP thread!
2022-01-05 15:48:48.845365 [INFO    ] [p1           ]  Found Configuration for 1
ports
2022-01-05 15:48:48.845563 [INFO    ] [syn1588      ]  Syn1588Ifc requires at least:
2022-01-05 15:48:48.845571 [INFO    ] [syn1588      ]  - linux driver version 1.4-15-
g05b7283
2022-01-05 15:48:48.845576 [INFO    ] [syn1588      ]  - windows driver version
10/05/2017, 10.9.16.
2022-01-05 15:48:48.845590 [INFO    ] [syn1588      ]  Device /dev/syncD0 found
2022-01-05 15:48:48.                  588           ]  syn1588(R) Hardware Clock M
2.3.5 f=125000000 Hz
2022-01-05 15:48:48.845617 [INFO    ] [syn1588      ]  Found stop clock support
2022-01-05 15:48:48.845630 [INFO    ] [syn1588      ]  Using MAC TS Version 3160
2022-01-05 15:48:48.845635 [INFO    ] [syn1588      ]  Using programmable 1-step TS
2022-01-05 15:48:48.845653                             syn1588(R) PCIe NIC Revision
2, Build 876
2022-01-05 15:48:48.845669                             Using syn1588 mode
2022-01-05 15:48:48.845699 [INFO    ] [p1.clock     ]  Spike M2S: Init with ival 0,
buffer size 16
2022-01-05 15:48:48.845762 [INFO    ] [p1.clock     ]  Spike Path: Init with ival 0,
buffer size 16
```

```
2022-01-05 15:48:48.845778 [INFO    ] [p1.io        ]  Init shared mem
2022-01-05 15:48:48.848090 [INFO    ] [syn1588      ]  syn1588HwClk: clearing leap
second jump
2022-01-05 15:48:48.848103 [INFO    ] [p1.engine    ]  Settings: ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848112 [INFO    ] [p1.engine    ]  Settings: Prio1 128 ClkClass
255 clkAccuracy 39 clkVariance 65535
2022-01-05 15:48:48.848119 [INFO    ] [p1.engine    ]  Settings: Prio2 128 Domain 0
2022-01-05 15:48:48.848235 [INFO    ] [p1.network.mcport]  SIOCSHWTSTAMP: tx_type 1
requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848245 [INFO    ] [p1.network.mcport]  Activated SO_TIMESTAMPING
hardware
2022-01-05 15:48:48.848340 [INFO    ] [p1.network.mcport]  SIOCSHWTSTAMP: tx_type 1
requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848349 [INFO    ] [p1.network.mcport]  Activated SO_TIMESTAMPING
hardware
2022-01-05 15:48:48.848599 [INFO    ] [p1.clock     ]  Clk: Using Oregano Systems;
syn1588(R) PCIe NIC Revision 2; 00:1E:C0:85:DE:2B
2022-01-05 15:48:48.848608 [INFO    ] [p1.clock     ]    with ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848742 [INFO    ] [p1.clock     ]  Clk: Resetting servos
2022-01-05 15:48:48.848748 [INFO    ] [p1.clock     ]  Clk: Resetting filters
```

figure 2 syn1588® PTP Stack: example log output verbosity level info

Wireshark

Wireshark is a network traffic and protocol recorder and analyzer. You can download this free software here: https://www.wireshark.org/

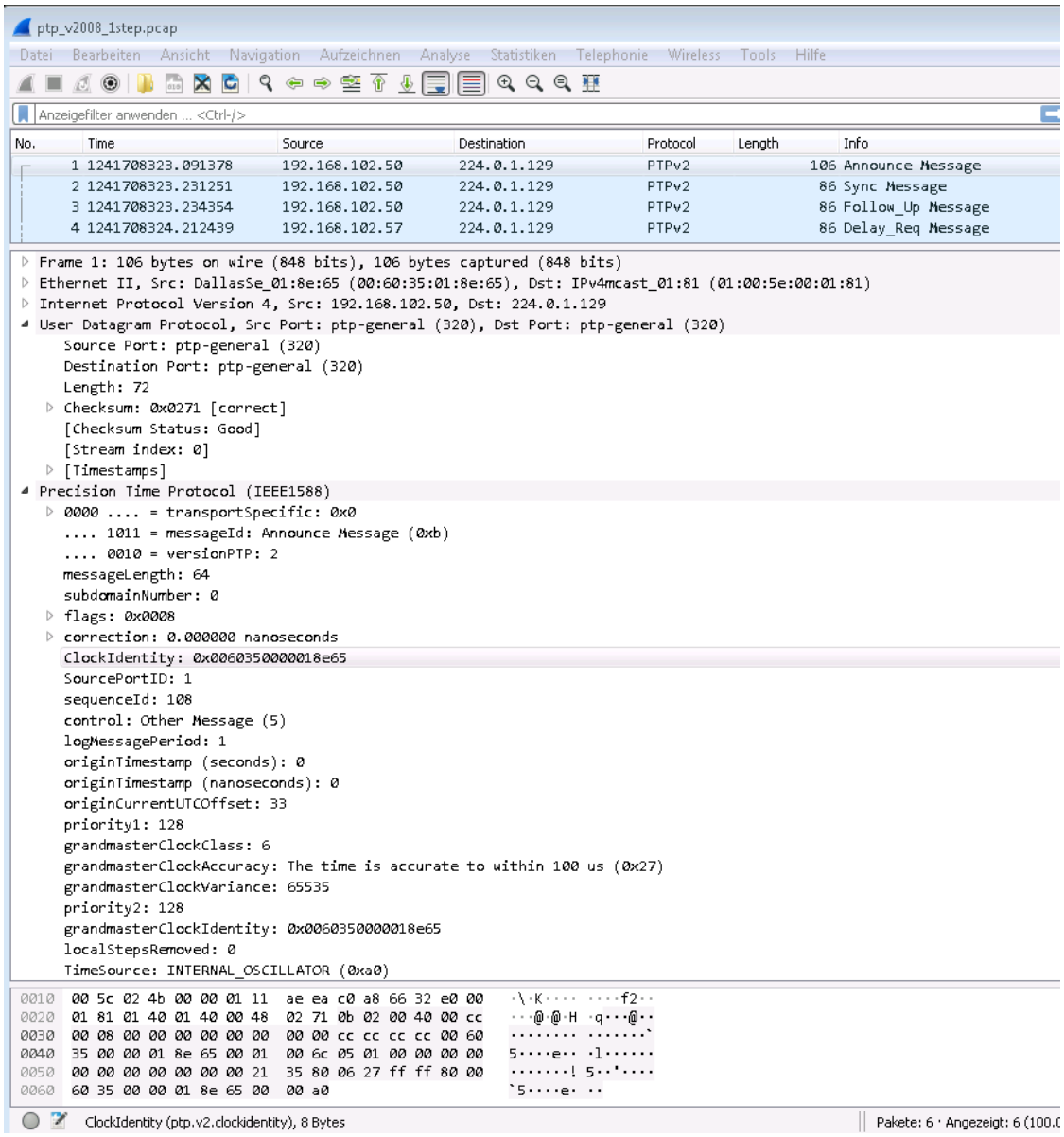

figure 3 Wireshark trace window

Analysis Procedure

The PTP communication (assuming one is using Layer 3 communication, thus UDP via either IPv4 or IPv6) requires a proper setup of the IP communication your system. This is nothing specific to PTP. You have to ensure a proper IP routing from your PTP node, using the selected network interface (typically a syn1588® PCIe NIC) to the Grandmaster. This basic IP setup shall be done using the standard operating system commands or procedures.

Check basic IP communication

One can use the PING command from a shell (DOS command window on Windows machines or any terminal on Linux machines) for this purpose. Simply ping the IP address of your PTP Grandmaster on the network.

```
[user@hugo a]$ ping 192.168.1.20
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data.
64 bytes from 192.168.1.20: icmp_seq=1 ttl=128 time=0.766 ms
64 bytes from 192.168.1.20: icmp_seq=2 ttl=128 time=0.490 ms
64 bytes from 192.168.1.20: icmp_seq=3 ttl=128 time=0.557 ms
64 bytes from 192.168.1.20: icmp_seq=4 ttl=128 time=0.509 ms
^C
--- 192.168.1.20 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3375ms
rtt min/avg/max/mdev = 0.490/0.580/0.766/0.112 ms
```

figure 4 Using PING to test basic IP communication

Caution

If your node owns several network interfaces you have to ensure proper IP routing. The PING test just gives you the information that there is a proper route from your node to the Grandmaster. It does not give you the information whether the network interface you plan to use for the PTP communication offers the required IP settings for proper IP communication.

Check PTP traffic on network interface

Now one can use Wireshark to check the incoming network traffic on the network interface one plans to use for PTP communication. One can simply filter the traffic for PTP messages only by entering the string “ptp” into the filtering window. Now, Wireshark displays all PTP messages seen on the selected network interface.

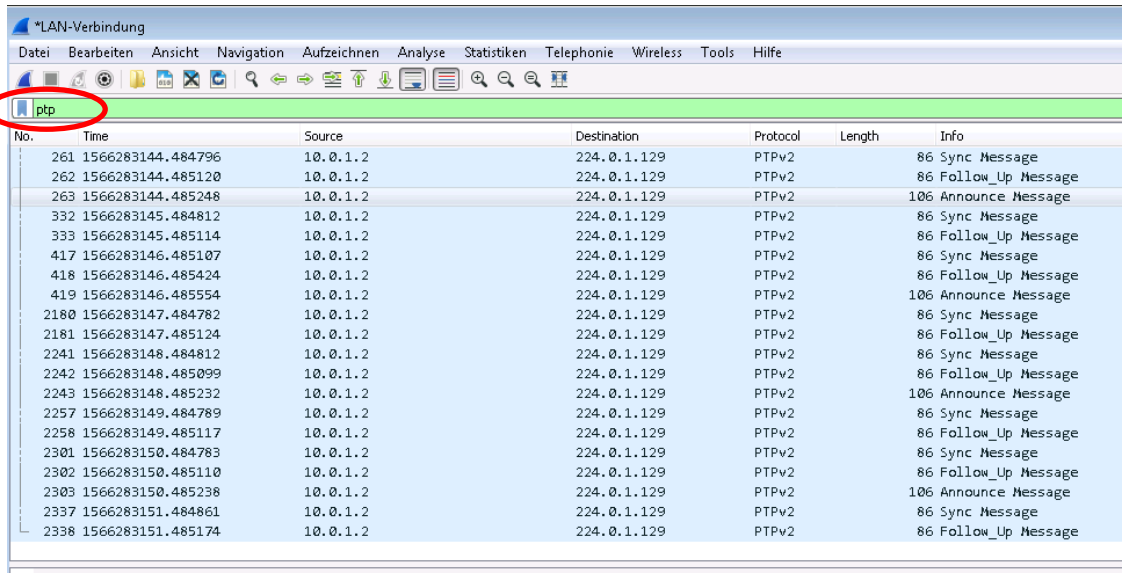

figure 5 Wireshark: check for PTP traffic

Caution

Wireshark is a network debugging tool. It does not care about proper IP settings nor on active firewalls or multicast routings on your node. You will see any message physically visible on the network interface although higher levels of your operating systems are instructed to block or re-route this traffic.

Check for syn1588® hardware

If one runs the syn1588® PTP Stack on a syn1588® PCIe NIC one can check whether the correct network interface had been chosen when invoking the syn1588® PTP Stack. The syn1588® PCIe NIC has to be reported in the log file as shown in the following example:

```
oregano@KAEFER:/opt/oregano/bin$ sudo ./ptp -i enp2s0 -C S -v info -r 0
2022-01-05 15:48:48.845172 [INFO    ] [             ]  syn1588(R) PTP Stack -
IEEE1588-2008 Engine
2022-01-05 15:48:48.845206 [INFO    ] [             ]  e: 2022-01-
05T12:43:16 - v1.14-9-g7cbf9e73
2022-01-05 15:48:48.845212 [INFO    ] [             ]  (c) Oregano Systems
- Design & Consulting GesmbH 2005-2022
2022-01-05 15:48:48.845217 [INFO    ] [             ]  Confidential unpublished data
- All rights reserved
2022-01-05 15:48:48.845224 [INFO    ] [             ]  syn1588(R) PTP Stack started:
2022-01-05 14:48:48.845222 (UTC)
2022-01-05 15:48:48.845317 [INFO    ] [p1           ]  Port 1: adding config "i" =
"enp2s0"
2022-01-05 15:48:48.845327 [INFO    ] [p1           ]  Port 1: adding config "C" =
"S"
2022-01-05 15:48:48.845333 [INFO    ] [p1           ]  Port 1: adding config "v" =
"info"
2022-01-05 15:48:48.845338 [INFO    ] [p1           ]  Port 1: adding config "r" =
"0"
2022-01-05 15:48:48.845344 [INFO    ] [p1           ]  PTP version 2.0
2022-01-05 15:48:48.845348 [INFO    ] [             ]  Command line: ./ptp -i enp2s0
-C S -v info -r 0
2022-01-05 15:48:48.845358 [WARNING ] [             ]  Failed to set high priority
for the PTP thread!
2022-01-05 15:48:48.845365 [INFO    ] [p1           ]  Found Configuration for 1
ports
2022-01-05 15:48:48.845563 [INFO    ] [syn1588      ]  Syn1588Ifc requires at least:
2022-01-05 15:48:48.845571 [INFO    ] [syn1588      ]  - linux driver version 1.4-15-
g05b7283
2022-01-05 15:48:48.845576 [INFO    ] [syn1588      ]  - windows driver version
10/05/2017, 10.9.16.182
2022-                                  l588]  Device /dev/syncD0 found
2022-01-05 15:48:48.845608 [INFO    ] [syn1588      ]  syn1588(R) Hardware Clock M
2.3.5 f=125000000 Hz
2022-01-05 15:48:48.845617 [INFO    ] [syn1588      ]  Found stop clock support
2022-01-05 15:48:48.845630 [INFO    ] [syn1588      ]  Using MAC TS Version 3160
2022-01-05 15:48:48.845635 [INFO    ] [syn1588      ]  Using programmable 1-step TS
2022-01-05 15:48:48                    8            ]  syn1588(R) PCIe NIC Revision
2, Build 876
2022-01-05 15:48:48                                 ]  Using syn1588 mode
2022-01-05 15:48:48.845699 [INFO    ] [p1.clock     ]  Spike M2S: Init with ival 0,
buffer size 16
2022-01-05 15:48:48.845762 [INFO    ] [p1.clock     ]  Spike Path: Init with ival 0,
buffer size 16
2022-01-05 15:48:48.845778 [INFO    ] [p1.io        ]  Init shared mem
2022-01-05 15:48:48.848090 [INFO    ] [syn1588      ]  syn1588HwClk: clearing leap
second jump
2022-01-05 15:48:48.848103 [INFO    ] [p1.engine    ]  Settings: ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848112 [INFO    ] [p1.engine    ]  Settings: Prio1 128 ClkClass
255 clkAccuracy 39 clkVariance 65535
```


figure 6 syn1588® PTP Stack: check for syn1588® hardware

If one runs the syn1588® PTP Stack on any network interface card no syn1588® hardware support will be found and reported as shown in the next example.

```
2022-01-07 13:24:24.758827 [INFO    ] [             ]  syn1588(R) PTP Stack -
IEEE1588-2008 Engine
2022-01-07 13:24:24.758888 [INFO    ] [             ]  Build date: 2021-12-
22T13:51:43 - v1.13-260-g3d9f6485
2022-01-07 13:24:24.758935 [INFO    ] [             ]  Command line: ./ptp -i enp3s0
-C S -v info -f /home/oregano/Documents/ptp_intelnic_hwmode.txt
2022-01-07 13:24:24.759002 [INFO    ] [             ]  syn1588(R) PTP Stack started:
2022-01-07 12:24:24.758995 (UTC)
2022-01-07 13:24:24.759050 [INFO    ] [             ]  Command line: ./ptp -i enp3s0
-C S -v info -f /home/oregano/Documents/ptp_intelnic_hwmode.txt
2022-01-07 13:24:24.759106 [WARNING ] [             ]  Failed to set high priority
for the PTP thread!
2022-01-07 13:24:24.759179 [INFO    ] [p1           ]  Found Configuration for 1
ports
2022-01-07 13:24:24.761645 [INFO    ] [syn1588      ]  Syn1588Ifc requires at least:
2022-01-07 13:24:24.761[illegible]                        driver version 1.4-15-
g05b7283
2022-01-07 13:24:24.761772 [INFO    ] [syn1588      ]  - windows driver version
10/05/2017, 10.9.16.182
2022-01-07 13:24:24.761947 [WARNING ] [p1           ]  There is no card with the
given ClockID available
2022-01-07 13:24:24.762095 [INFO    ] [p1           ]  Device /dev/ptp1 found
2022-01-07 13:24:24.762168 [INFO    ] [p1           ]  Using PHC mode

2022-01-07 13:24:24.762311 [INFO    ] [p1.clock     ]  Spike M2S: Init with ival 0,
buffer size 16
2022-01-07 13:24:24.762718 [INFO    ] [p1.clock     ]  Spike Path: Init with ival 0,
buffer size 16
2022-01-07 13:24:24.762789 [INFO    ] [p1.io        ]  Init shared mem
2022-01-07 13:24:24.767247 [INFO    ] [p1.engine    ]  Settings: ClockId
68:05:ca:ff:fe:2c:da:ca
2022-01-07 13:24:24.767327 [INFO    ] [p1.engine    ]  Settings: Prio1 128 ClkClass
255 clkAccuracy 39 clkVariance 65535
2022-01-07 13:24:24.767366 [INFO    ] [p1.engine    ]  Settings: Prio2 128 Domain 0
2022-01-07 13:24:24.768093 [INFO    ] [p1.network.mcport]  SIOCSHWTSTAMP: tx_type 1
requested, got 1; rx_filter 0 requested, got 12
2022-01-07 13:24:24.768171 [INFO    ] [p1.network.mcport]  Activated SO_TIMESTAMPING
hardware
2022-01-07 13:24:24.768804 [INFO    ] [p1.network.mcport]  SIOCSHWTSTAMP: tx_type 1
requested, got 1; rx_filter 0 requested, got 12
2022-01-07 13:24:24.768880 [INFO    ] [p1.network.mcport]  Activated SO_TIMESTAMPING
hardware
2022-01-07 13:24:2[illegible]                             ]  Clk: Using Linux PHC;;
2022-01-07 13:24:2[illegible]                             ]    with ClockId
68:05:ca:ff:fe:2c:da:ca
2022-01-07 13:24:24.771834 [INFO    ] [p1.clock     ]  Clk: Resetting servos
2022-01-07 13:24:24.771941 [INFO    ] [p1.clock     ]  Clk: Resetting filters
```


figure 7 syn1588® PTP Stack: without syn1588® hardware (but with Linux PHC support)

Check for ANNOUNCE messages

After successfully starting the syn1588® PTP Stack on the intended network interface one shall receive ANNOUNCE messages from the PTP Grandmaster. For this run the syn1588® PTP Stack at least with log level 3. All incoming, received PTP messages are flagged by an text arrow to the right “--->”. One can simply grep for these lines in the text output.

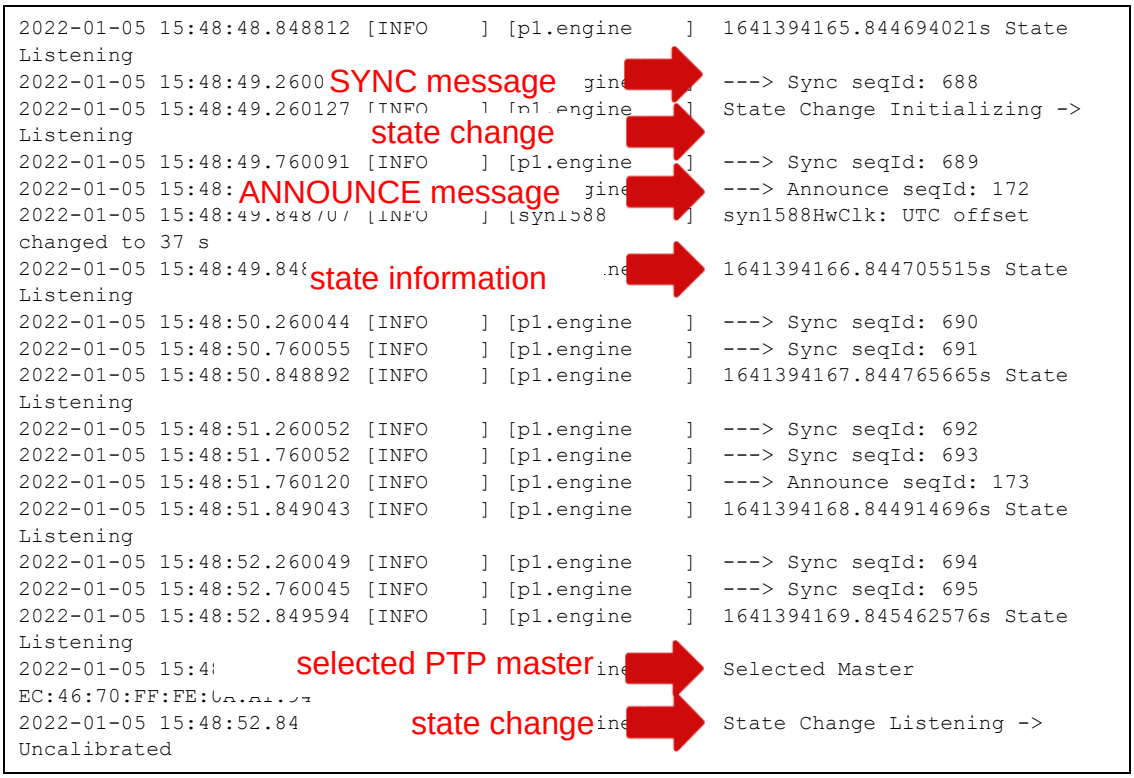

figure 8 syn1588® PTP Stack: check for incoming ANNOUNCE messages

In this example a SYNC messages is received first. The local PTP node now moves from the state “Initializing” to the state “Listening”. Without receiving SYNC or ANNOUNCE messages the node will remain in the state “Initializing”. If one does not see ANNOUNCE messages here your PTP, IP or node setup is corrupted. Either an improper IP configuration may inhibit the syn1588® PTP Stack to receive the Grandmaster’s PTP messages or a local firewall blocks the PTP traffic. PTP utilize the UDP ports 319 and 320. Correct your system settings until you see the expected PTP messages here. Additionally, the Grandmaster may use another PTP clock domain than it is configured for the PTP slave. One can use Wireshark and check the PTP clock domain used by the Grandmaster.

Note, that the announce interval of the PTP slaves must match the announce interval used by the Grandmaster. The PTP slaves derive an announce timeout period based on the configured announce interval. If the

Grandmaster sends ANNOUNCE messages not within the announce timeout period the PTP slave will also never change its state to “Slave”.

After receiving twice the correct ANNOUNCE message the PTP slave moves from the state “Listening” to the state “Uncalibrated”; the PTP master had been accepted now.

One can use Wireshark on the same network interface to check whether the expected ANNOUNCE messages are physically available on the network interface.

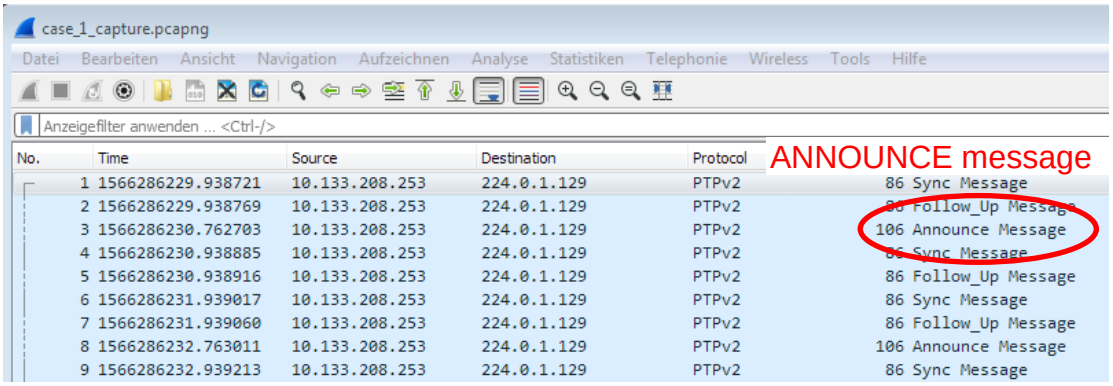

figure 9 Wireshark: check for incoming ANNOUNCE messages

If one would like to check the PTP clock domain used by the Grandmaster, select e.g. an Announce message sent by the Grandmaster and check the PTP message details as shown in the next figure.

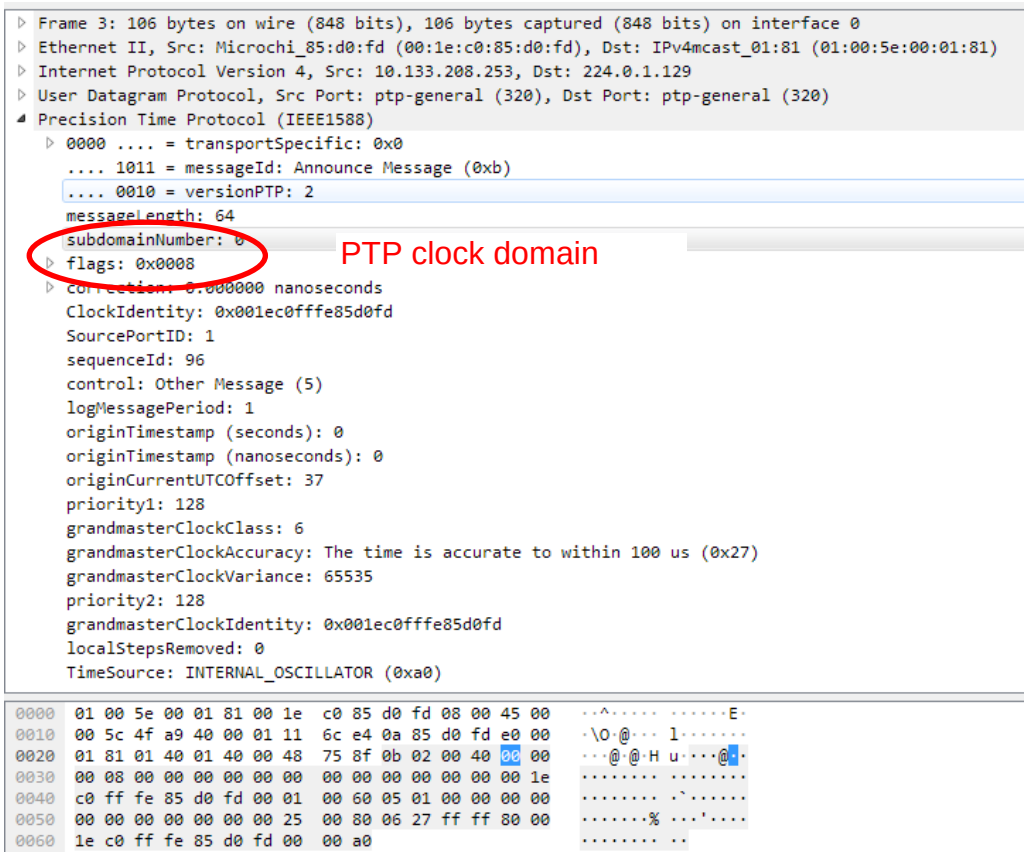

figure 10 Wireshark: check for the PTP clock domain

The following syn1588® PTP Stack log output shows an error condition. This happens if a local firewall blocks the PTP traffic on the PTP slave. No SYNC and ANNOUNCE messages can be received by the syn1588® PTP Stack.

```
oregano@KAEFER:/opt/oregano/bin$ sudo ./ptp -i enp2s0 -C S -v info -r 0
2022-01-05 15:48:48.845172 [INFO    ] [             ]  syn1588(R) PTP Stack -
IEEE1588-2008 Engine
2022-01-05 15:48:48.845206 [INFO    ] [             ]  Build date: 2022-01-
05T12:43:16 - v1.14-9-g7cbf9e73
2022-01-05 15:48:48.845212 [INFO    ] [             ]  Copyright (c) Oregano Systems
- Design & Consulting GesmbH 2005-2022
2022-01-05 15:48:48.845217 [INFO    ] [             ]  Confidential unpublished data
- All rights reserved
2022-01-05 15:48:48.845224 [INFO    ] [             ]  syn1588(R) PTP Stack started:
2022-01-05 14:48:48.845222 (UTC)
2022-01-05 15:48:48.845317 [INFO    ] [p1           ]  Port 1: adding config "i" =
"enp2s0"
2022-01-05 15:48:48.845327 [INFO    ] [p1           ]  Port 1: adding config "C" =
"S"
2022-01-05 15:48:48.845333 [INFO    ] [p1           ]  Port 1: adding config "v" =
"info"
2022-01-05 15:48:48.845338 [INFO    ] [p1           ]  Port 1: adding config "r" =
"0"
2022-01-05 15:48:48.845344 [INFO    ] [p1           ]  PTP version 2.0
2022-01-05 15:48:48.845348 [INFO    ] [             ]  Command line: ./ptp -i enp2s0
-C S -v info -r 0
2022-01-05 15:48:48.845358 [WARNING ] [             ]  Failed to set high priority
for the PTP thread!
2022-01-05 15:48:48.845365 [INFO    ] [p1           ]  Found Configuration for 1
ports
2022-01-05 15:48:48.845563 [INFO    ] [syn1588      ]  Syn1588Ifc requires at least:
2022-01-05 15:48:48.845571 [INFO    ] [syn1588      ]  - linux driver version 1.4-15-
g05b7283
2022-01-05 15:48:48.845576 [INFO    ] [syn1588      ]  - windows driver version
10/05/2017, 10.9.16.182
2022-01-05 15:48:48.845590 [INFO    ] [syn1588      ]  Device /dev/syncD0 found
2022-01-05 15:48:48.845608 [INFO    ] [syn1588      ]  syn1588(R) Hardware Clock M
2.3.5 f=125000000 Hz
2022-01-05 15:48:48.845617 [INFO    ] [syn1588      ]  Found stop clock support
2022-01-05 15:48:48.845630 [INFO    ] [syn1588      ]  Using MAC TS Version 3160
2022-01-05 15:48:48.845635 [INFO    ] [syn1588      ]  Using programmable 1-step TS
2022-01-05 15:48:48.845653 [INFO    ] [syn1588      ]  syn1588(R) PCIe NIC Revision
2, Build 876
2022-01-05 15:48:48.845669 [INFO    ] [p1           ]  Using syn1588 mode
2022-01-05 15:48:48.845699 [INFO    ] [p1.clock     ]  Spike M2S: Init with ival 0,
buffer size 16
2022-01-05 15:48:48.845762 [INFO    ] [p1.clock     ]  Spike Path: Init with ival 0,
buffer size 16
2022-01-05 15:48:48.845778 [INFO    ] [p1.io        ]  Init shared mem
2022-01-05 15:48:48.848090 [INFO    ] [syn1588      ]  syn1588HwClk: clearing leap
second jump
2022-01-05 15:48:48.848103 [INFO    ] [p1.engine    ]  Settings: ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848112 [INFO    ] [p1.engine    ]  Settings: Prio1 128 ClkClass
255 clkAccuracy 39 clkVariance 65535
2022-01-05 15:48:48.848119 [INFO    ] [p1.engine    ]  Settings: Prio2 128 Domain 0
2022-01-05 15:48:48.848235 [INFO    ] [p1.network.mcport]  SIOCSHWTSTAMP: tx_type 1
requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848245 [INFO    ] [p1.network.mcport]  Activated SO_TIMESTAMPING
hardware
2022-01-05 15:48:48.848340 [INFO    ] [p1.network.mcport]  SIOCSHWTSTAMP: tx_type 1
requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848349 [INFO    ] [p1.network.mcport]  Activated SO_TIMESTAMPING
hardware
```

no incoming messages and no outgoing messages

```
2022-01-05 15:48:48.848599 [INFO    ] [p1.clock     ]  Clk: Using Oregano Systems;
syn1588(R) PCIe NIC Revision 2; 00:1E:C0:85:DE:2B
2022-01-05 15:48:48.848608 [INFO    ] [p1.clock     ]    with ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848742 [INFO    ] [p1.clock     ]  Clk: Resetting servos
2022-01-05 15:48:48.848748 [INFO    ] [p1.clock     ]  Clk: Resetting filters
2022-01-05 15:48:48.848755 [INFO    ] [p1.clock     ]  Spike M2S: Init with ival 0,
buffer size 16
2022-01-05 15:48:48.848764 [INFO    ] [p1.clock     ]  Spike Path: Init with ival 0,
buffer size 16
2022-01-05 15:48:48.848774 [INFO    ] [p1.io        ]  Init shared mem
2022-01-05 15:48:48.848787 [INFO    ] [syn1588      ]  syn1588HwClk: clearing leap
second jump
2022-01-05 15:48:48.848812 [INFO                    state Listening 4165.844694021s State
Listening
2022-01-05 15:48:49.848829 [INFO    ] [p1.engine    ]  1641394166.844705515s State
Listening
2022-01-05 15:48:50.848892 [INFO    ] [p1.engine    ]  1641394167.844765665s State
Listening
2022-01-05 15:48:51.849043 [INFO    ] [p1.engine    ]  1641394168.844914696s State
Listening
```

figure 11 syn1588® PTP Stack: active firewall blocking PTP traffic

Note, there are no incoming messages flagged by “--->” seen in the log file. The PTP slave remains in the state “Listening”. Please further note, that the syn1588® PTP Stack does not send any packet in this state since it instructed to act as PTP slave; there are no outgoing messages as well “<---”. In Wireshark one can still see these PTP messages on the network interface. Wireshark does not care about firewalls.

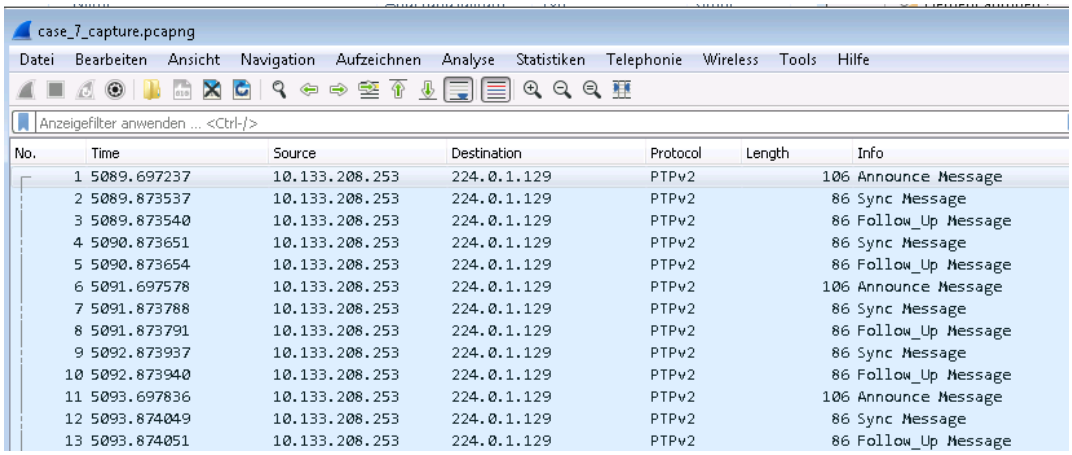

figure 12 Wireshark: PTP messages with active firewall blocking PTP traffic

The following syn1588® PTP Stack log output shows what happens if the PTP slave receives messages from

If the Grandmaster sends PTP messages with a wrong PTP clock domain these messages have to be ignored by the syn1588® PTP Stack. Thus the PTP slave remains in the state “Listening”. If one increases the syn1588® PTP Stack’ log level using the maximum verbosity level of 4 one will see a hint for such a wrong PTP clock domain value.

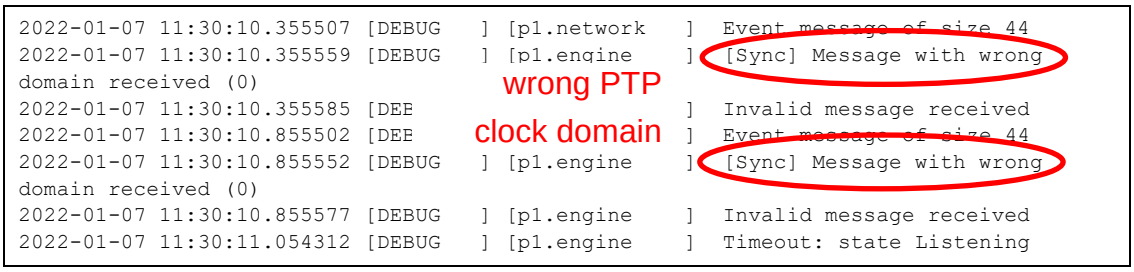

figure 13 syn1588® PTP Stack: wrong clock domain with log level debug

Check for SYNC messages

One can use the identical procedure as described in the previous chapter to check for SYNC messages received. The following excerpt of the syn1588® PTP Stack log file shows the received SYNC messages.

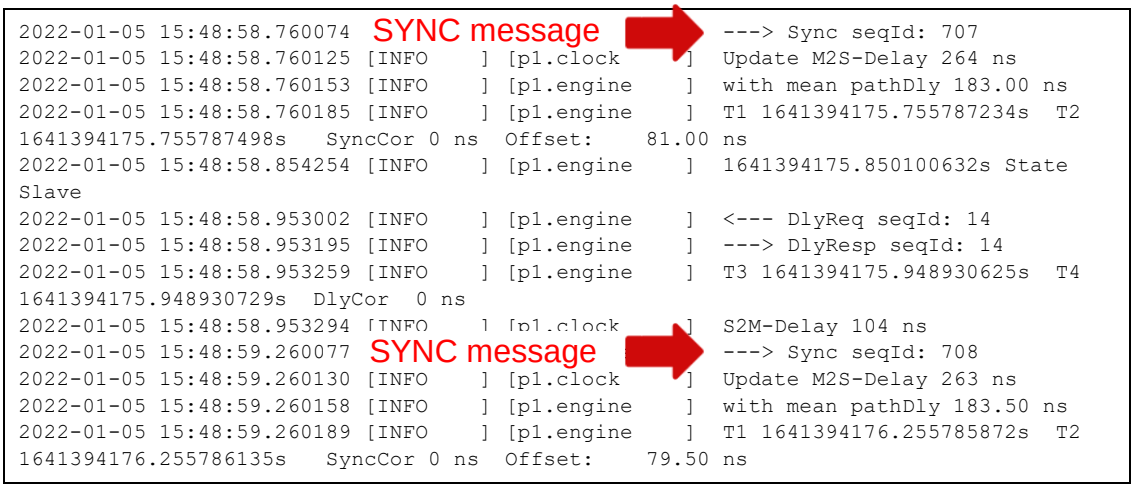

figure 14 syn1588® PTP Stack: check for incoming SYNC messages

PTP defines two basic communication mechanisms.

- 2-step mode
- 1-step mode

In the example shown above 1-step mode is used. The sender inserts the timestamp on-the-fly directly into the SYNC packet. For 2-step operation every SYNC is accompanied by a FOLLOW-UP message that sends the timestamp

Check for DELAY REQUEST & DELAY RESPONSE messages

As soon as the PTP slave proceeds in the “Uncalibrated” state it starts sending DELAY REQUEST messages that shall be answered by DELAY RESPONSE message by the Grandmaster. Note, that the DELAY REQUEST messages are sent (outgoing message) by the PTP slave and thus are preceded by a arrow to the left “<---“. One can again grep for this pattern.

```
2022-01-05 15:48:57.760077 [INFO    ] [p1.engine    ]  ---> Sync seqId: 705
2022-01-05 15:48:57.760129 [INFO    ] [p1.clock     ]  Update M2S-Delay 274 ns
2022-01-05 15:48:57.760156 [INFO    ] [p1.clock     ]  Spike M2S: Init with ival -1,
buffer size 32
2022-01-05 15:48:57.760195 [INFO    ] [p1.engine    ]  with mean pathDly 189.50 ns
2022-01-05 15:48:57.760219 [INFO    ] [p1.engine    ]  T1 1641394174.755789711s  T2
1641394174.755789985s   SyncCor 0 ns  Offset:    84.50 ns
2022-01-05 15:48:57.760277 [INFO    ] [p1.engine    ]  ---> Announce seqId: 176
2022-01-05 15:48:57.853340 [INFO    ] [p1.engine    ]  1641394174.849190507s State
Slave
2022-01-05 15:48:58.086576 [INFO    ] [p1.engine    ]  <--- DlyReq seqId: 12
2022-01-05 15:48:58.086775 [INFO    ] [p1.engine    ]  ---> DlyResp seqId: 12
2022-01-05 15:48:58.086812 [INFO    ] [p1.engine    ]  T3 1641394175.082511583s  T4
1641394175.082511679s  DlyCor  0 ns
2022-01-05 15:48:58.086875 [INFO    ] [p1.clock     ]  S2M-Delay 96 ns
```

DELAY REQUEST message

DELAY RESPONSE message

figure 15 syn1588® PTP Stack: check for outgoing DELAY REQUEST and incoming DELAY RESONSE messages

If one does not receive DELAY RESPONSE messages the DELAY REQUEST messages might have been filtered on their way to the Grandmaster.

Transition to Slave State

After successfully receiving several DELAY RESPONSE messages the PTP slave proceeds from the state “Uncalibrated” to the state “Slave”.

```
2022-01-05 15:48:57.260080 [INFO    ] [p1.engine    ]  ---> Sync seqId: 704
2022-01-05 15:48:57.260124 [INFO    ] [p1.clock     ]  Update M2S-Delay 277 ns
2022-01-05 15:48:57.260164 [INFO    ] [p1.engine    ]  with mean pathDly 189.50 ns
2022-01-05 15:48:57.260193 [INFO    ] [p1.engine    ]  T1 1641394174.255790991s  T2
1641394174.255791268s   SyncCor 0 ns  Offset:    87.50 ns
2022-01-05 15:48:57.486784 [INFO    ] [p1.clock     ]  DriftCalc: drift is 5 ns/s
2022-01-05 15:48:57.486845 [INFO    ] [syn1588      ]  Adjust rate by -4184.9 ns/s
(speeding up)
2022-01-05 15:48:57.486893 [INFO    ] [syn1588      ]  Setting step to 0x00000800,
0x0231b033, 8.000033479 ns
2022-01-05 15:48:57.486944 [INFO    ] [p1.clock     ]  Clk: Resetting filters
2022-01-05 15:48:57.486969 [INFO    ] [p1.clock     ]  Spike M2S: Init with ival 0,
buffer size 16
2022-01-05 15:48:57.487039 [INFO    ] [p1.clock     ]  Spike Path: Init with ival 0,
buffer size 16
2022-01-05 15:48:57.487118 [INFO    ] [p1.clock     ]  Drift calc completed
2022-01-05 15:48:57.487164 [INFO    ] [p1.engine    ]  <---  DlyReq seqId: 10
2022-01-05 15:48:57.487298 [INFO    ] [p1.engine    ]  State Change Uncalibrated ->
Slave
2022-01-05 15:48:57.487419 [INFO    ] [p1.engine    ]  ---> DlyResp seqId: 10
2022-01-05 15:48:57.487446 [INFO    ] [p1.engine    ]  T3 1641394174.483103322s  T4
1641394174.483103421s  DlyCor  0 ns
2022-01-05 15:48:57.487471 [INFO    ] [p1.clock     ]  S2M-Delay 99 ns
2022-01-05 15:48:57.487497 [INFO    ] [p1.clock     ]  Spike Path: Init with ival -1,
buffer size 32
2022-01-05 15:48:57.741356 [INFO    ] [p1.engine    ]  <--- DlyReq seqId: 11
2022-01-05 15:48:57.741552 [INFO    ] [p1.engine    ]  ---> DlyResp seqId: 11
2022-01-05 15:48:57.741593 [INFO    ] [p1.engine    ]  T3 1641394174.737306786s  T4
1641394174.737306891s  DlyCor  0 ns
2022-01-05 15:48:57.741627 [INFO    ] [p1.clock     ]  S2M-Delay 105 ns
2022-01-05 15:48:57.760077 [INFO    ] [p1.engine    ]  ---> Sync seqId: 705
2022-01-05 15:48:57.760129 [INFO    ] [p1.clock     ]  Update M2S-Delay 274 ns
2022-01-05 15:48:57.760156 [INFO    ] [p1.clock     ]  Spike M2S: Init with ival -1,
buffer size 32
2022-01-05 15:48:57.760195 [INFO    ] [p1.engine    ]  with mean pathDly 189.50 ns
2022-01-05 15:48:57.760219 [INFO    ] [p1.engine    ]  T1 1641394174.755789711s  T2
1641394174.755789985s   SyncCor 0 ns  Offset:    84.50 ns
2022-01-05 15:48:57.760277 [INFO    ] [p1.engine    ]  ---> Announce seqId: 176
2022-01-05 15:48:57.853340 [INFO    ] [p1.engine    ]  T4 1641394174.849190507s State
Slave
2022-01-05 15:48:58.086576 [INFO    ] [p1.engine    ]  <--- DlyReq seqId: 12
2022-01-05 15:48:58.086775 [INFO    ] [p1.engine    ]  ---> DlyResp seqId: 12
2022-01-05 15:48:58.086839 [INFO    ] [p1.engine    ]  T3 1641394175.082511583s  T4
1641394175.082511679s  DlyCor  0 ns
2022-01-05 15:48:58.086875 [INFO    ] [p1.clock     ]  S2M-Delay 96 ns
```


figure 16 syn1588® PTP Stack: transition to state “Slave”

Now the PTP slave is fully synchronized with the master. The syn1588® PTP Stack log output also reports the measured delays Master-to-Slave (M2S) and slave-to-Master (S2M) as well as the timestamps T1 – T4.

Offset to Master

One shall observe that over the time the offset of the slave to the master is reduced to a minimum. The speed of this adaptation depends on the selected clock servo algorithm and parameters. The syn1588® PTP Stack log output displays two types of offset values:

- unfiltered offset
- filtered offset

Only the latter value is a useful measure for the current state of the synchronization.

```
2022-01-05 15:49:03.260086 [INFO    ] [p1.engine    ]  ---> Sync seqId: 716
2022-01-05 15:49:03.260138 [INFO    ] [p1.clock     ]  Update M2S-Delay 173 ns
2022-01-05 15:49:03.260167 [INFO    ] [p1.engine    ]  with mean pathDly 189.00 ns
2022-01-05 15:49:03.260197 [INFO    ] [p1.engine    ]  T1 1641394180.255775631s  T2
1641394180.255775804s   SyncCor 0 ns  Offset:   -16.00 ns
2022-01-05 15:49:03.557410 [INFO    ] [p1.engine    ]  <--- DlyReq seqId: 22
2022-01-05 15:49:03.557584 [INFO    ] [p1.engine    ]  ---> DlyResp seqId: 22
2022-01-05 15:49:03.557623 [INFO    ] [p1.engine    ]  T3 1641394180.553316087s  T4
1641394180.553316299s  DlyCor  0 ns
2022-01-05 15:49:03.557667 [INFO    ] [p1.clock     ]  S2M-Delay 212 ns
2022-01-05 15:49:03.557694 [INFO    ] [p1.clock     ]  Spike Path: Ignoring 212
2022-01-05 15:49:03.760095 [INFO    ] [p1.engine    ]  ---> Sync seqId: 717
2022-01-05 15:49:03.760146 [INFO    ] [p1.clock     ]  Update M2S-Delay 169 ns
2022-01-05 15:49:03.760174 [INFO    ] [p1.engine    ]  with mean pathDly 184.00 ns
2022-01-05 15:49:03.760205 [INFO    ] [p1.engine    ]  T1 1641394180.755775656s  T2
1641394180.755775825s   SyncCor 0 ns  Offset:   -15.00 ns
2022-01-05 15:49:03.760257 [INFO    ] [p1.engine    ]  ---> Announce seqId: 179
2022-01-05 15:49:03.857178 [INFO    ] [p1.engine    ]  1641394180.853000946s State
Slave
2022-01-05 15:49:03.857271 [INFO    ] [p1.clock     ]  Adjusting clock at -15.00 ns
offset
2022-01-05 15:49:03.857425 [INFO    ] [p1.clock     ]  changed boundary to       25
ns(17)
2022-01-05 15:49:03.857469 [INFO    ] [p1.clock     ]  Spike M2S: mean: 171, median:
172, variance: 66
2022-01-05 15:49:03.857578 [INFO    ] [p1.clock     ]  Spike Path: mean: 202, median:
202, variance: 114
2022-01-05 15:49:03.857603 [INFO    ] [syn1588      ]  Adjust rate by -4188.77 ns/s
(speeding up)
2022-01-05 15:49:03.857712 [INFO    ] [syn1588      ]  Setting step to 0x00000800,
0x023234f3, 8.000033510 ns
2022-01-05 15:49:04.191773 [INFO    ] [p1.engine    ]  <--- DlyReq seqId: 23
2022-01-05 15:49:04.192077 [INFO    ] [p1.engine    ]  ---> DlyResp seqId: 23
2022-01-05 15:49:04.192124 [INFO    ] [p1.engine    ]  T3 1641394181.187676695s  T4
1641394181.187676904s  DlyCor  0 ns
2022-01-05 15:49:04.192154 [INFO    ] [p1.clock     ]  S2M-Delay 209 ns
```


figure 17 syn1588® PTP Stack: offset to the master

Multicast Operation

When using multicast network mode for PTP the whole (!) network infrastructure has to be set up to properly connect also the multicast messages from the Grandmaster to all PTP nodes.

Presence of a Transparent Clock

In case transparent clocks (TCs) are on the path from the Grandmaster to the PTP slave another field will be displayed in the syn1588® PTP Stack log output:

- the correction field

The correction field value summarizes all time the PTP packet resided onto such a transparent clock while packet traverses from the Grandmaster to the PTP slave and vice versa. The syn1588® PTP Stack uses this information in it's offset calculation. If no standard Ethernet switches are used in the network but just TCs, the resulting master-to-slave offset is almost the same as if the Grandmaster and PTP slave would have been connected by a direct network cable.

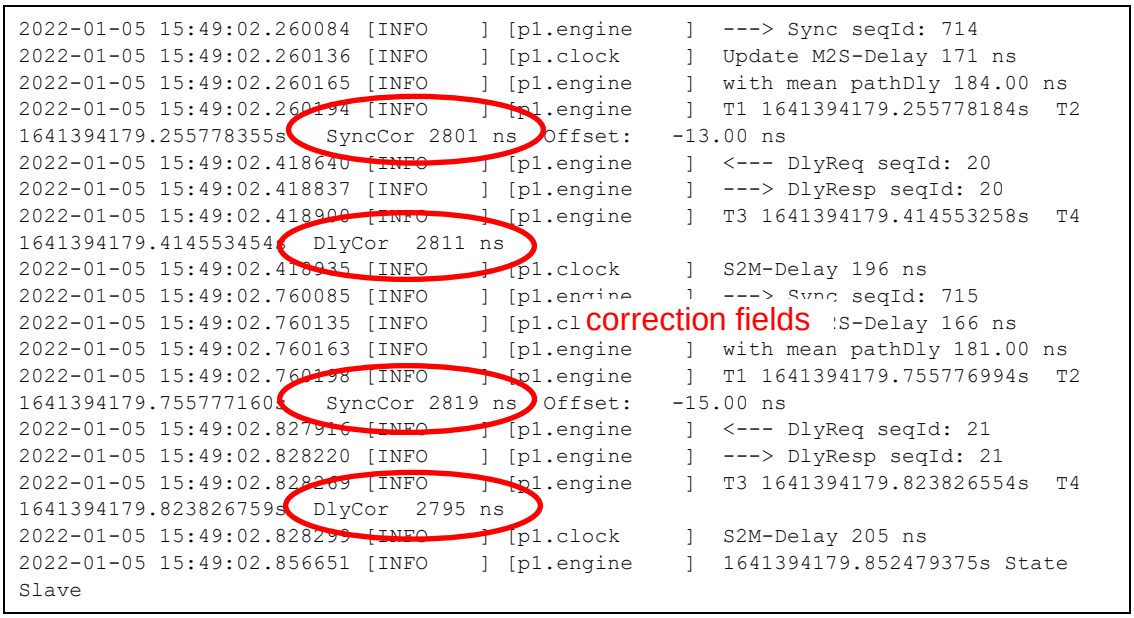

figure 18 syn1588® PTP Stack: presence of a TC – correction fields

Literature

AN004. (Version 1.6 - May 2019). *Application Note: "syn1588® PCIe NIC - Quick Start Guide".* Oregano Systems.

AN005. (Version 1.2 - March 2019). *Application Note: "syn1588® System Information Report".* Oregano Systems.

Version 1.3 - January 2022

Oregano Systems
A Meinberg Company

Franzosengraben 8
A-1030 Vienna
Austria
http://oregano.at
contact@oregano.at