

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.

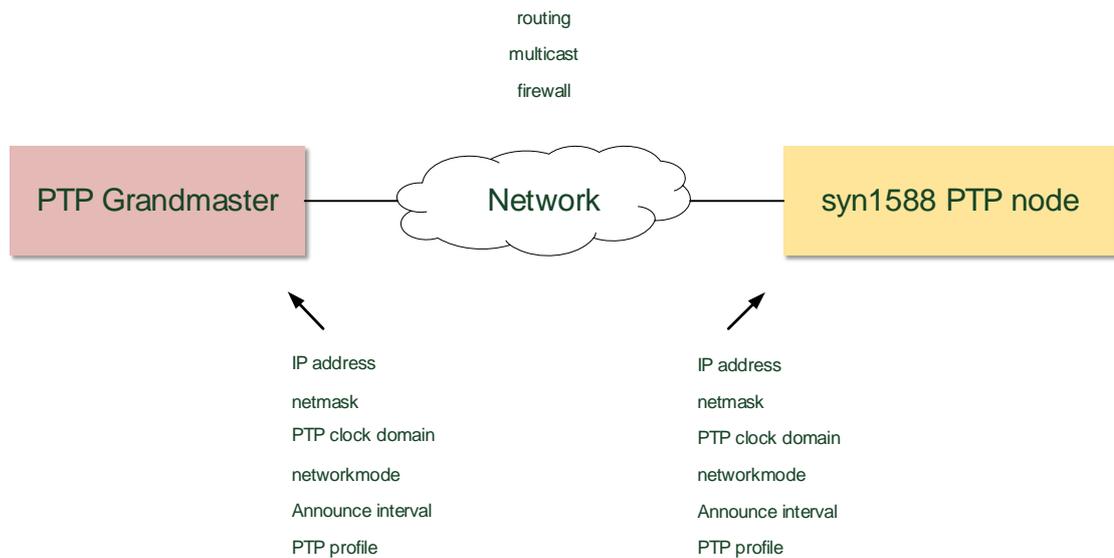


figure 1 Application scenario

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 ] [ ] software e: 2022-01-
05T12:43:16 - v1.14-9-g7cbf9e73 ← build information
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2022-01-05 15:48:48.845217 [INFO ] [ ] Confidential unpublished data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 15:48:48.845590 [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 ← syn1588® hardware
2022-01-05 15:48:48.845669 [INFO ] [ ] 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    ] [pl.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    ] [pl.engine     ] Settings: ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848112 [INFO    ] [pl.engine     ] Settings: Prio1 128 ClkClass
255 clkAccuracy 39 clkVariance 65535
2022-01-05 15:48:48.848119 [INFO    ] [pl.engine     ] Settings: Prio2 128 Domain 0
2022-01-05 15:48:48.848235 [INFO    ] [pl.network.mcport] SIOCSHWTSTAMP: tx_type 1
requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848245 [INFO    ] [pl.network.mcport] Activated SO_TIMESTAMPING
hardware
2022-01-05 15:48:48.848340 [INFO    ] [pl.network.mcport] SIOCSHWTSTAMP: tx_type 1
requested, got 1; rx_filter 0 requested, got 12
2022-01-05 15:48:48.848349 [INFO    ] [pl.network.mcport] Activated SO_TIMESTAMPING
hardware
2022-01-05 15:48:48.848599 [INFO    ] [pl.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    ] [pl.clock      ]      with ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848742 [INFO    ] [pl.clock      ] Clk: Resetting servos
2022-01-05 15:48:48.848748 [INFO    ] [pl.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/>

ptp_v2008_1step.pcap

Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telephonie Wireless Tools Hilfe

Anzeigefilter anwenden ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	1241708323.091378	192.168.102.50	224.0.1.129	PTPv2	106	Announce Message
2	1241708323.231251	192.168.102.50	224.0.1.129	PTPv2	86	Sync Message
3	1241708323.234354	192.168.102.50	224.0.1.129	PTPv2	86	Follow_Up Message
4	1241708324.212439	192.168.102.57	224.0.1.129	PTPv2	86	Delay_Req Message

▶ Frame 1: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
 ▶ Ethernet II, Src: DallasSe_01:8e:65 (00:60:35:01:8e:65), Dst: IPv4mcast_01:81 (01:00:5e:00:01:81)
 ▶ Internet Protocol Version 4, Src: 192.168.102.50, Dst: 224.0.1.129
 ▶ User Datagram Protocol, Src Port: ptp-general (320), Dst Port: ptp-general (320)
 Source Port: ptp-general (320)
 Destination Port: ptp-general (320)
 Length: 72
 ▶ Checksum: 0x0271 [correct]
 [Checksum Status: Good]
 [Stream index: 0]
 ▶ [Timestamps]

▶ Precision Time Protocol (IEEE1588)
 ▶ 0000 = transportSpecific: 0x0
 ... 1011 = messageId: Announce Message (0xb)
 ... 0010 = versionPTP: 2
 messageLength: 64
 subdomainNumber: 0
 ▶ flags: 0x0008
 ▶ correction: 0.000000 nanoseconds
 ClockIdentity: 0x0060350000018e65
 SourcePortID: 1
 sequenceId: 108
 control: Other Message (5)
 logMessagePeriod: 1
 originTimestamp (seconds): 0
 originTimestamp (nanoseconds): 0
 originCurrentUTCOffset: 33
 priority1: 128
 grandmasterClockClass: 6
 grandmasterClockAccuracy: The time is accurate to within 100 us (0x27)
 grandmasterClockVariance: 65535
 priority2: 128
 grandmasterClockIdentity: 0x0060350000018e65
 localStepsRemoved: 0
 TimeSource: INTERNAL_OSCILLATOR (0xa0)

```

0010  00 5c 02 4b 00 00 01 11 ae ea c0 a8 66 32 e0 00  ·\·K····· ····f2··
0020  01 81 01 40 01 40 00 48 02 71 0b 02 00 40 00 cc  ··@·@·H·q···@··
0030  00 08 00 00 00 00 00 00 00 00 cc cc cc cc 00 60  ······@·······
0040  35 00 00 01 8e 65 00 01 00 6c 05 01 00 00 00 00  5····e···1······
0050  00 00 00 00 00 00 00 21 35 80 06 27 ff ff 80 00  ······!5··'····
0060  60 35 00 00 01 8e 65 00 00 a0  ·5····e···
  
```

ClockIdentity (ptp.v2.clockidentity), 8 Bytes

Pakete: 6 · Angezeigt: 6 (100%)

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
```

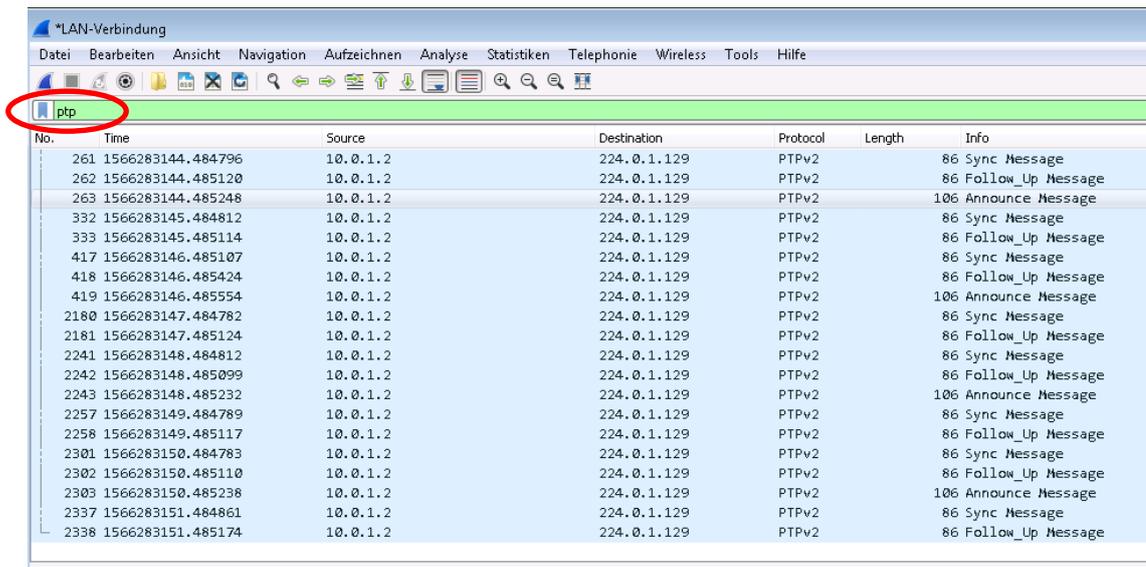
figure 4 Using PING to test basic IP communication

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.



No.	Time	Source	Destination	Protocol	Length	Info
261	1566283144.484796	10.0.1.2	224.0.1.129	PTPv2	86	Sync Message
262	1566283144.485120	10.0.1.2	224.0.1.129	PTPv2	86	Follow-Up Message
263	1566283144.485248	10.0.1.2	224.0.1.129	PTPv2	106	Announce Message
332	1566283145.484812	10.0.1.2	224.0.1.129	PTPv2	86	Sync Message
333	1566283145.485114	10.0.1.2	224.0.1.129	PTPv2	86	Follow-Up Message
417	1566283146.485107	10.0.1.2	224.0.1.129	PTPv2	86	Sync Message
418	1566283146.485424	10.0.1.2	224.0.1.129	PTPv2	86	Follow-Up Message
419	1566283146.485554	10.0.1.2	224.0.1.129	PTPv2	106	Announce Message
2180	1566283147.484782	10.0.1.2	224.0.1.129	PTPv2	86	Sync Message
2181	1566283147.485124	10.0.1.2	224.0.1.129	PTPv2	86	Follow-Up Message
2241	1566283148.484812	10.0.1.2	224.0.1.129	PTPv2	86	Sync Message
2242	1566283148.485099	10.0.1.2	224.0.1.129	PTPv2	86	Follow-Up Message
2243	1566283148.485232	10.0.1.2	224.0.1.129	PTPv2	106	Announce Message
2257	1566283149.484789	10.0.1.2	224.0.1.129	PTPv2	86	Sync Message
2258	1566283149.485117	10.0.1.2	224.0.1.129	PTPv2	86	Follow-Up Message
2301	1566283150.484783	10.0.1.2	224.0.1.129	PTPv2	86	Sync Message
2302	1566283150.485110	10.0.1.2	224.0.1.129	PTPv2	86	Follow-Up Message
2303	1566283150.485238	10.0.1.2	224.0.1.129	PTPv2	106	Announce Message
2337	1566283151.484861	10.0.1.2	224.0.1.129	PTPv2	86	Sync Message
2338	1566283151.485174	10.0.1.2	224.0.1.129	PTPv2	86	Follow-Up Message

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 ] [ ] software build information 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-
syn1588® hardware found
2022-01-05 15:48:48.845608 [INFO ] [syn1588 ] Device /dev/syncD0 found
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 syn1588® hardware 8 ] syn1588(R) PCIe NIC Revision
2, Build 876
syn1588® hardware build information
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 ClockID derived from MAC address
2022-01-05 15:48:48.848112 [INFO ] [p1.engine ] Settings: Priol 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 13:24:24.761705 [INFO ] [syn1588 ] 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 13:24:24.771115 [INFO ] [p1.clock ] Clk: Using Linux PHC;;
2022-01-07 13:24:24.771115 [INFO ] [p1.clock ] 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.

```

2022-01-05 15:48:48.848812 [INFO    ] [pl.engine  ] 1641394165.844694021s State
Listening
2022-01-05 15:48:49.2600 [INFO    ] [pl.engine  ] ---> Sync seqId: 688
2022-01-05 15:48:49.260127 [INFO    ] [pl.engine  ] State Change Initializing ->
Listening
2022-01-05 15:48:49.760091 [INFO    ] [pl.engine  ] ---> Sync seqId: 689
2022-01-05 15:48:49.848807 [INFO    ] [pl.engine  ] ---> Announce seqId: 172
2022-01-05 15:48:49.848807 [INFO    ] [syn1588    ] syn1588HwClk: UTC offset
changed to 37 s
2022-01-05 15:48:49.848807 [INFO    ] [pl.engine  ] 1641394166.844705515s State
Listening
2022-01-05 15:48:50.260044 [INFO    ] [pl.engine  ] ---> Sync seqId: 690
2022-01-05 15:48:50.760055 [INFO    ] [pl.engine  ] ---> Sync seqId: 691
2022-01-05 15:48:50.848892 [INFO    ] [pl.engine  ] 1641394167.844765665s State
Listening
2022-01-05 15:48:51.260052 [INFO    ] [pl.engine  ] ---> Sync seqId: 692
2022-01-05 15:48:51.760052 [INFO    ] [pl.engine  ] ---> Sync seqId: 693
2022-01-05 15:48:51.760120 [INFO    ] [pl.engine  ] ---> Announce seqId: 173
2022-01-05 15:48:51.849043 [INFO    ] [pl.engine  ] 1641394168.844914696s State
Listening
2022-01-05 15:48:52.260049 [INFO    ] [pl.engine  ] ---> Sync seqId: 694
2022-01-05 15:48:52.760045 [INFO    ] [pl.engine  ] ---> Sync seqId: 695
2022-01-05 15:48:52.849594 [INFO    ] [pl.engine  ] 1641394169.845462576s State
Listening
2022-01-05 15:48:52.849594 [INFO    ] [pl.engine  ] Selected Master
EC:46:70:FF:FE:0A:11:04
2022-01-05 15:48:52.849594 [INFO    ] [pl.engine  ] State Change Listening ->
Uncalibrated

```

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

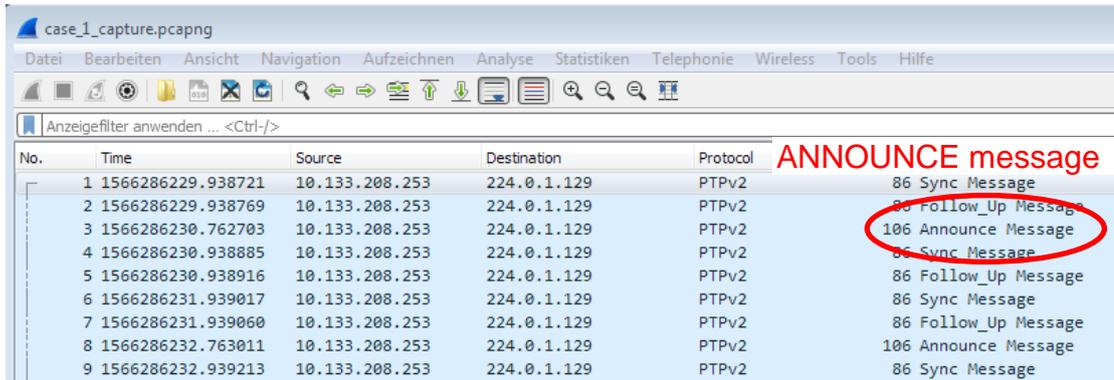
In this example a SYNC messages is received first. The local PTP node now moves from the state “Initializing” to the state “Listening”. Without receiving SYNC or ANNOUNCE messages the node will remain in the state “Initializing”. If one does not see ANNOUNCE messages here your PTP, IP or node setup is 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.



No.	Time	Source	Destination	Protocol	Message
1	1566286229.938721	10.133.208.253	224.0.1.129	PTPv2	86 Sync Message
2	1566286229.938769	10.133.208.253	224.0.1.129	PTPv2	86 Follow-Up Message
3	1566286230.762703	10.133.208.253	224.0.1.129	PTPv2	106 Announce Message
4	1566286230.938885	10.133.208.253	224.0.1.129	PTPv2	86 Sync Message
5	1566286230.938916	10.133.208.253	224.0.1.129	PTPv2	86 Follow-Up Message
6	1566286231.939017	10.133.208.253	224.0.1.129	PTPv2	86 Sync Message
7	1566286231.939060	10.133.208.253	224.0.1.129	PTPv2	86 Follow-Up Message
8	1566286232.763011	10.133.208.253	224.0.1.129	PTPv2	106 Announce Message
9	1566286232.939213	10.133.208.253	224.0.1.129	PTPv2	86 Sync Message

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.

```

▶ Frame 3: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
▶ Ethernet II, Src: Microchi_85:d0:fd (00:1e:c0:85:d0:fd), Dst: IPv4mcast_01:81 (01:00:5e:00:01:81)
▶ Internet Protocol Version 4, Src: 10.133.208.253, Dst: 224.0.1.129
▶ User Datagram Protocol, Src Port: ptp-general (320), Dst Port: ptp-general (320)
▲ Precision Time Protocol (IEEE1588)
  ▶ 0000 .... = transportSpecific: 0x0
    .... 1011 = messageId: Announce Message (0xb)
    .... 0010 = versionPTP: 2
    messageLength: 64
    subdomainNumber: 0
    flags: 0x0008
    correction: 0.000000 nanoseconds
    ClockIdentity: 0x001ec0fffe85d0fd
    SourcePortID: 1
    sequenceId: 96
    control: Other Message (5)
    logMessagePeriod: 1
    originTimestamp (seconds): 0
    originTimestamp (nanoseconds): 0
    originCurrentUTCOffset: 37
    priority1: 128
    grandmasterClockClass: 6
    grandmasterClockAccuracy: The time is accurate to within 100 us (0x27)
    grandmasterClockVariance: 65535
    priority2: 128
    grandmasterClockIdentity: 0x001ec0fffe85d0fd
    localStepsRemoved: 0
    TimeSource: INTERNAL_OSCILLATOR (0xa0)

```

PTP clock domain

```

0000  01 00 5e 00 01 81 00 1e c0 85 d0 fd 08 00 45 00  ..^.....E-
0010  00 5c 4f a9 40 00 01 11 6c e4 0a 85 d0 fd e0 00  ..\O.@...l.....
0020  01 81 01 40 01 40 00 48 75 8f 0b 02 00 40 00 00  ..:@.Hu...@.
0030  00 08 00 00 00 00 00 00 00 00 00 00 00 00 1e  .....
0040  c0 ff fe 85 d0 fd 00 01 00 60 05 01 00 00 00 00  .....
0050  00 00 00 00 00 00 00 25 00 80 06 27 ff ff 80 00  .....% .....
0060  1e c0 ff fe 85 d0 fd 00 00 a0  .....

```

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] SIOCSHWSTAMP: 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] SIOCSHWSTAMP: 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 ] [pl.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 ] [pl.clock ] with ClockId
00:1e:c0:ff:fe:85:de:2b
2022-01-05 15:48:48.848742 [INFO ] [pl.clock ] Clk: Resetting servos
2022-01-05 15:48:48.848748 [INFO ] [pl.clock ] Clk: Resetting filters
2022-01-05 15:48:48.848755 [INFO ] [pl.clock ] Spike M2S: Init with ival 0,
buffer size 16
2022-01-05 15:48:48.848764 [INFO ] [pl.clock ] Spike Path: Init with ival 0,
buffer size 16
2022-01-05 15:48:48.848774 [INFO ] [pl.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 ] [pl.engine ] 1641394166.844705515s State
Listening
2022-01-05 15:48:50.848892 [INFO ] [pl.engine ] 1641394167.844765665s State
Listening
2022-01-05 15:48:51.849043 [INFO ] [pl.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.

No.	Time	Source	Destination	Protocol	Length	Info
1	5089.697237	10.133.208.253	224.0.1.129	PTPv2		106 Announce Message
2	5089.873537	10.133.208.253	224.0.1.129	PTPv2		86 Sync Message
3	5089.873540	10.133.208.253	224.0.1.129	PTPv2		86 Follow_Up Message
4	5090.873651	10.133.208.253	224.0.1.129	PTPv2		86 Sync Message
5	5090.873654	10.133.208.253	224.0.1.129	PTPv2		86 Follow_Up Message
6	5091.697578	10.133.208.253	224.0.1.129	PTPv2		106 Announce Message
7	5091.873788	10.133.208.253	224.0.1.129	PTPv2		86 Sync Message
8	5091.873791	10.133.208.253	224.0.1.129	PTPv2		86 Follow_Up Message
9	5092.873937	10.133.208.253	224.0.1.129	PTPv2		86 Sync Message
10	5092.873940	10.133.208.253	224.0.1.129	PTPv2		86 Follow_Up Message
11	5093.697836	10.133.208.253	224.0.1.129	PTPv2		106 Announce Message
12	5093.874049	10.133.208.253	224.0.1.129	PTPv2		86 Sync Message
13	5093.874051	10.133.208.253	224.0.1.129	PTPv2		86 Follow_Up Message

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.

```

2022-01-07 11:30:10.355507 [DEBUG ] [p1.network ] Event message of size 44
2022-01-07 11:30:10.355559 [DEBUG ] [p1.engine ] [Sync] Message with wrong
domain received (0)
2022-01-07 11:30:10.355585 [DEE ] [p1.engine ] Invalid message received
2022-01-07 11:30:10.855502 [DEE ] [p1.engine ] Event message of size 44
2022-01-07 11:30:10.855552 [DEBUG ] [p1.engine ] [Sync] Message with wrong
domain received (0)
2022-01-07 11:30:10.855577 [DEBUG ] [p1.engine ] Invalid message received
2022-01-07 11:30:11.054312 [DEBUG ] [p1.engine ] Timeout: state Listening

```

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.

```

2022-01-05 15:48:58.760074 SYNC message ---> Sync seqId: 707
2022-01-05 15:48:58.760125 [INFO ] [p1.clock ] Update M2S-Delay 264 ns
2022-01-05 15:48:58.760153 [INFO ] [p1.engine ] with mean pathDly 183.00 ns
2022-01-05 15:48:58.760185 [INFO ] [p1.engine ] T1 1641394175.755787234s T2
1641394175.755787498s SyncCor 0 ns Offset: 81.00 ns
2022-01-05 15:48:58.854254 [INFO ] [p1.engine ] 1641394175.850100632s State
Slave
2022-01-05 15:48:58.953002 [INFO ] [p1.engine ] <--- DlyReq seqId: 14
2022-01-05 15:48:58.953195 [INFO ] [p1.engine ] ---> DlyResp seqId: 14
2022-01-05 15:48:58.953259 [INFO ] [p1.engine ] T3 1641394175.948930625s T4
1641394175.948930729s DlyCor 0 ns
2022-01-05 15:48:58.953294 [INFO ] [p1.clock ] S2M-Delay 104 ns
2022-01-05 15:48:59.260077 SYNC message ---> Sync seqId: 708
2022-01-05 15:48:59.260130 [INFO ] [p1.clock ] Update M2S-Delay 263 ns
2022-01-05 15:48:59.260158 [INFO ] [p1.engine ] with mean pathDly 183.50 ns
2022-01-05 15:48:59.260189 [INFO ] [p1.engine ] T1 1641394176.255785872s T2
1641394176.255786135s SyncCor 0 ns Offset: 79.50 ns

```

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 DELAY REQUEST message
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 15:48:58.086875 [INFO    ] [p1.clock   ] T3 1641394175.082511583s T4
1641394175.082511679s DlyCor 0 ns
2022-01-05 15:48:58.086875 [INFO    ] [p1.clock   ] S2M-Delay 96 ns

```

figure 15 syn1588® PTP Stack: check for outgoing DELAY REQUEST and incoming DELAY RESPONSE 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 15:48:57.260114 [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 15:48:57.487497 [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 ] 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 unfiltered offset
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 filtered 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.

```

2022-01-05 15:49:02.260084 [INFO ] [p1.engine ] ---> Sync seqId: 714
2022-01-05 15:49:02.260136 [INFO ] [p1.clock ] Update M2S-Delay 171 ns
2022-01-05 15:49:02.260165 [INFO ] [p1.engine ] with mean pathDly 184.00 ns
2022-01-05 15:49:02.260194 [INFO ] [p1.engine ] T1 1641394179.255778184s T2
1641394179.255778355s SyncCor 2801 ns Offset: -13.00 ns
2022-01-05 15:49:02.418640 [INFO ] [p1.engine ] <--- DlyReq seqId: 20
2022-01-05 15:49:02.418837 [INFO ] [p1.engine ] ---> DlyResp seqId: 20
2022-01-05 15:49:02.418900 [INFO ] [p1.engine ] T3 1641394179.414553258s T4
1641394179.414553454s DlyCor 2811 ns
2022-01-05 15:49:02.418935 [INFO ] [p1.clock ] S2M-Delay 196 ns
2022-01-05 15:49:02.760085 [INFO ] [p1.engine ] ---> Sync seqId: 715
2022-01-05 15:49:02.760135 [INFO ] [p1.cl correction fields :S-Delay 166 ns
2022-01-05 15:49:02.760163 [INFO ] [p1.engine ] with mean pathDly 181.00 ns
2022-01-05 15:49:02.760198 [INFO ] [p1.engine ] T1 1641394179.755776994s T2
1641394179.755777160s SyncCor 2819 ns Offset: -15.00 ns
2022-01-05 15:49:02.827916 [INFO ] [p1.engine ] <--- DlyReq seqId: 21
2022-01-05 15:49:02.828220 [INFO ] [p1.engine ] ---> DlyResp seqId: 21
2022-01-05 15:49:02.828269 [INFO ] [p1.engine ] T3 1641394179.823826554s T4
1641394179.823826759s DlyCor 2795 ns
2022-01-05 15:49:02.828299 [INFO ] [p1.clock ] S2M-Delay 205 ns
2022-01-05 15:49:02.856651 [INFO ] [p1.engine ] 1641394179.852479375s State
Slave

```

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.



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

Copyright © 2022

Oregano Systems – Design & Consulting GmbH

ALL RIGHTS RESERVED.

Oregano Systems does not assume any liability arising out of the application or use of any product described or shown herein nor does it convey any license under its patents, copyrights, or any rights of others.

Licenses or any other rights such as, but not limited to, patents, utility models, trademarks or tradenames, are neither granted nor conveyed by this document, nor does this document constitute any obligation of the disclosing party to grant or convey such rights to the receiving party.

Oregano Systems reserves the right to make changes, at any time without notice, in order to improve reliability, function or design. Oregano Systems will not assume responsibility for the use of any circuitry described herein.

All trademarks used in this document are the property of their respective owners.