



Abstract

redSync is another utility of the syn1588® software suite. It allows the redundant synchronization of a PTP Slave to two PTP Grandmasters. redSync is based on the syn1588® PTP Stack as well as the syn1588® hardware. The redSync utility is available starting with the release Q3/2020 (v1.12) of the syn1588® software suite. This application note describes the basic usage of this utility.

Primary assumption for operation

The redundancy provided by redSync relies on the following assumption:

Both PTP Grandmasters use the same main time source. This is typically GPS. They may use different connections to this time source, e.g., different GPS receivers.

redSync will provide seamless switchover, IFF both PTP Grandmasters provide the same/similar time, i.e., within 100 ns is recommended. This is typically the case for GPS synchronized PTP Masters. Higher differences between the two PTP Grandmasters will result in delayed switchover.

Scenario A – Hardware Clock Synchronization

The basic application structure for using redSync to provide redundancy for a single hardware clock synchronization is as follows.

- Two independent instances of the syn1588[®] PTP Stack connect to one PTP Grandmaster each
- A single syn1588[®] PCIe NIC that connects to the network over which the two PTP Grandmasters can be accessed

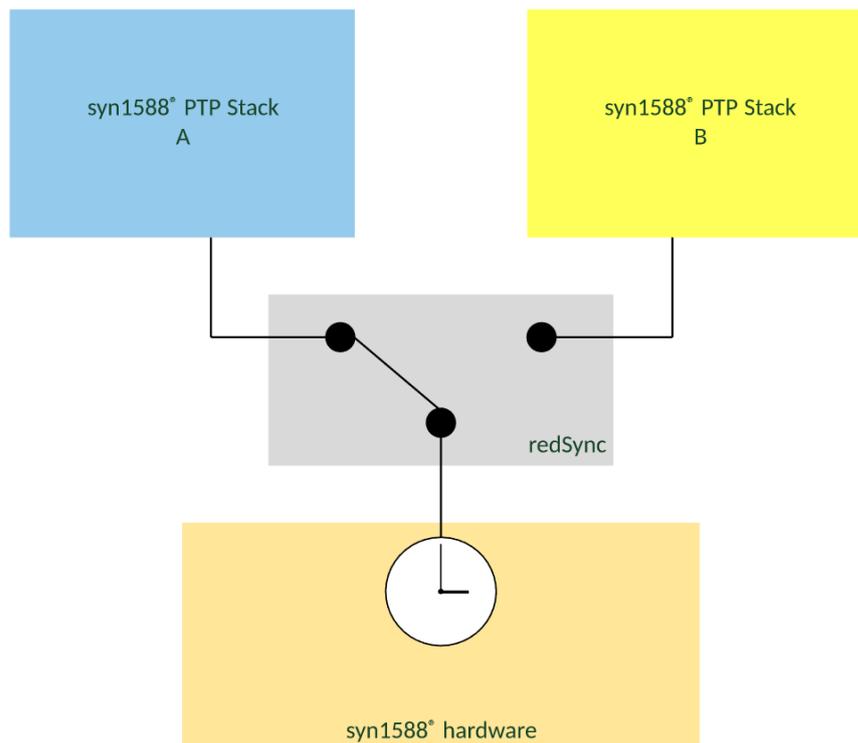


Figure 1: redSync for redundant Hardware Clock Synchronization

redSync observes the synchronization status of these two syn1588[®] PTP Stack instances and selects on a set of configurable criteria the syn1588[®] PTP Stack instance that actually shall maintain and control the high-precision PTP hardware clock. If, for example, the connection of the active syn1588[®] PTP Stack instance to its PTP Grandmaster is lost or dramatically degraded the second syn1588[®] PTP Stack instance will take over seamlessly without deteriorating the accuracy.

Scenario B – System Clock Synchronization

The basic application structure for using redSync to provide redundancy for System Clock synchronization is as follows.

- Two independent instances of the syn1588[®] PTP Stack connect to one PTP Grandmaster each
- Each syn1588[®] PTP Stack uses a syn1588[®] PCIe NIC or another IEEE1588 capable network interface
- A degraded operation is also possible with one or both syn1588[®] PTP Stack instances operating in software mode
- The syn1588[®] ISync utility is configured for operation together with redSync
- redSync selects the best PTP grandmaster from the two PTP instances and forwards this information to ISync
- ISync uses the selected PTP instance (and the synchronized hardware clock) to synchronize the system clock

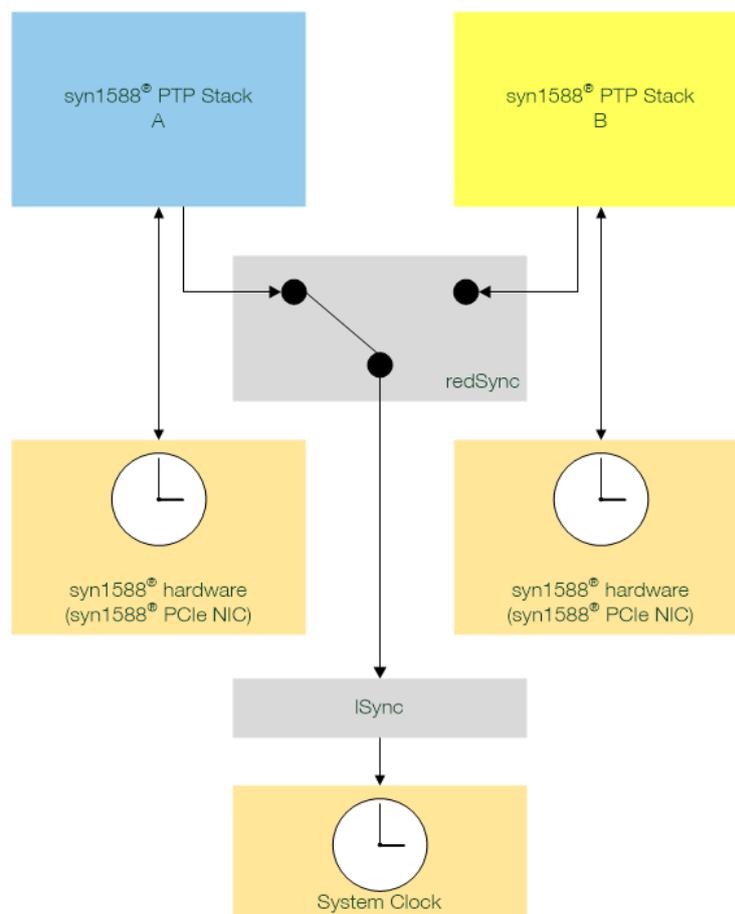


Figure 2: redSync for redundant System Clock Synchronization

Current Support

Note that, redSync is currently supported on Linux OS.
Windows support Has not been fully qualified but is possible.

Layer 2 support is currently not supported.

The supported and tested PTP profiles for redSync are:

- default
- SMPTE (2059.1)
- telecom2 (G.8275.1v2.1)
- G8275_2_s
- enterprise

Support for other PTP profiles will be added step-by-step with the following releases of the syn1588[®] software suite.

Pre-requisites

Scenario A

The following pre-requisites are required to redundantly synchronize a PTP node using redSync:

- PTP node with syn1588[®] hardware offering two independent timestamping units on the same physical network interface
- PTP node with Linux OS (Windows support not fully qualified)
- two syn1588[®] PTP Stack instances bound to the same (!) physical network interface and thus to the same PTP hardware clock
- each of the two syn1588[®] PTP Stack instances have to use an independent timestamping unit
- Both syn1588[®] PTP Stack instances have to run in PTP slave mode
- Both syn1588[®] PTP Stack instances have to use the identical network mode and PTP profile

Scenario B

The following pre-requisites are required to redundantly synchronize a PTP node using redSync:

- PTP node with two syn1588[®] PCIe NICs
OR
with two IEEE 1588 capable NICs
- PTP node with Linux OS (Windows support not fully qualified)
- two syn1588[®] PTP Stack instances each bound to one of the two NICs
- Both syn1588[®] PTP Stack instances have to run in PTP slave mode
- The syn1588[®] PTP Stack instances may use different network modes and PTP profiles
- IPv4 and Linux OS: Both syn1588 PTP Stacks will need to use different IPv4 subnets or VLAN

Important: The current selection modes “simple” and “bmc” do not incorporate the check for a synchronization accuracy boundary. Use a setup with high message rates ($\geq 8/s$) for Sync and DelayReq/Resp messages for a seamless switchover in case of re-establishment of the old, best PTP master.

Configuration

redSync connects to the two syn1588[®] PTP Stack instances specified on the command line via their respective PTP PortId = ClockID + Port Number. Depending on the selected decision algorithm redSync decides which of the two syn1588[®] PTP Stack shall be used as time source and thus actually synchronize the node’s clock. This selection is done via the Shared Memory interface. If redSync initiates a switch from one syn1588[®] PTP Stack instance to another the clock control implemented in the syn1588[®] PTP Stacks or the syn1588[®] ISync utility will guarantee a seamless switchover, IFF the primary assumption for redSync operation is satisfied!

redSync may be started in a restrictive “monitored” mode, which will stop operation if certain criteria are not met or in “unmonitored” mode, which will only log unmet criteria.

redSync may be started with different selection modes. “Simple” uses the first port (Port A) by default as source and only switches to the second port (Port B) if the first is no longer available. “BMC” applies the PTP Best Master Clock algorithm as criteria to select between the two PTP Stacks.

Table 1: Mode overview

Decision mode (compare method) [-m simple bmc]	simple (default)	redSync sets the first port (port A) active as long as it is alive and in slave state, otherwise the second (port B).	
	bmc	redSync decides based on the IEEE1588 best master clock algorithm (BMCA)	
Environment mode (environment state) [-e unmonitored monitored]	monitored strict condition checks	start	Both ports must be alive and in slave state.
		runtime	At least one port must be alive and in slave state. Otherwise redSync gives warning and terminates
	unmonitored (default) loose checks	start	redSync starts in all conditions and simply waits for a port alive and in slave state to set it as active port
		runtime	never terminates

Parameter: Monitored mode (-e monitored)

In this mode redSync expects that the system on which it is operated is monitored by a supervisor. For that case redSync expects strict start conditions to be fulfilled. If any of them is not met it will not start and return information about the problem. With this information the supervisor can and has to decide what needs to be done to bring the system into an operational state. redSync will also monitor these conditions throughout the operation and will terminate if both redundant PTP ports violate the conditions at the same time.

These conditions are

- syn1588[®] PTP stack running for both PTP ports.
- Both PTP ports are in slave state.

Parameter: Unmonitored mode (-e unmonitored)

redSync will not enforce the above mentioned conditions. It will start as well as continue to run even if these are violated. Violation of the conditions will only be reported in the log output.

Parameter: Simple Mode (-s simple)

In this mode redSync is instructed to choose the first given PTP port A as default.

If the connection to the PTP Grandmaster or the synchronization to the PTP Grandmaster fails for PTP port A, redSync will switch to PTP port B which is now allowed the control and maintain the PTP hardware clock. PTP port B will remain the active control port unless PTP port A synchronizes again to a PTP Grandmaster. redSync will switch back in such an event to PTP port A. PTP port B actually acts as a backup connection for PTP port A in this mode of operation.

Parameter: BMC Mode (-s bmc)

The BMC algorithm is normally used by the syn1588[®] PTP Stack to determine which PTP Grandmaster in a PTP clock domain shall be chosen. redSync uses this decision algorithm to decide between two PTP Grandmasters that are active in different PTP clock domains. redSync decides based on the Grandmaster properties and will allow the PTP port connected to the better PTP Grandmaster to be used as source. If the active PTP port loses the connection to its Grandmaster or the Grandmaster's performance is degraded the other PTP port will be selected.

Parameter: Port A (-A <clockID,portNumber>) and Port B (-B <>)

These two parameters are mandatory and set the two ports that are used for redundant synchronization.

A port is identified by its Port ID, as specified by the IEEE1588 standard. Hence, a clockID and a port number have to be given as value to this parameter.

For example:

```
"-A 8c:a5:a1:ff:fe:00:10:20,1 -B 8c:a5:a1:ff:fe:33:33:33,1"
```

This will select a syn1588[®] PTP Stack that is currently controlling the PTP Clock identified by its clockID 8c:a5:a1:ff:fe:00:10:20 and connected to the PTP port number 1 for Port A. Whereas Port B is identified by a different clock ID = 8c:a5:a1:ff:fe:33:33:33 and port number 1.

This example setup is a basic configuration for a redSync application suited for Scenario B, i.e., using two syn1588[®] interfaces in parallel (different clockIDs).

Parameter: System Clock Synchronization (-l)

This flag parameter (takes no value) has to be used to setup redSync for choosing the source to synchronize the system clock = Scenario B. If this parameter is not set, redSync will chose the source for synchronizing the hardware clock = Scenario A.

Parameter: Verbosity (-v 1|2|3|4)

This parameter can be used to get more verbose log output. A higher number will increase the verbosity level of redSync.

Parameter: help (-h)

This will print additional help for redSync usage into the shell.

Example – Scenario A

The setup of the system is as depicted in Figure 1.

Two syn1588[®] PTP Stack instances have to be invoked in the background and have to synchronize to two different PTP grandmaster instances over the network.

A single syn1588[®] PCIe NIC is required for this setup.

The different programs should be invoked in the following order.

syn1588[®] PCIe NIC

For this example a syn1588 PCIe NIC is mounted in the PC and named “enp1s0”.

The MAC address of this NIC is: 8c:a5:a1:00:00:c0 and the clockID for PTP operation is derived from it.

clockID: 8c:a5:a1:ff:fe:00:00:c0

syn1588[®] PTP Stack for Port A

```
> ptp ./portA.cfg
```

Where the PTP Stack configuration file contains the following:

```
port1:  
interface enp1s0  
domain 50
```

... to synchronize to a PTP Grandmaster (PTP domain: 50) over the local Ethernet interface “enp1s0”

This PTP instance can be identified by its clockID (derived from the interface “enp1s0” and the port number.

PortID: 8c:a5:a1:ff:fe:00:00:c0, 1

syn1588® PTP Stack for Port B

```
> ptp ./portB.cfg
```

Where the PTP Stack configuration file contains the following:

```
port1:
interface 127.0.0.1
port2:
interface enp1s0
domain 100
timestamper userTS
```

... to synchronize to a PTP Grandmaster (PTP domain: 100) over the same local Ethernet interface “enp1s0” and using a different timestamper unit (userTS) for PTP Packet timestamps.

Note that “port1” links to the local loopback to create a dummy port for the PTP Stack for the second instance. The PTP Stack instance for Port B is actually setup to operate as “port2”.

This PTP instance can be identified by its clockID (derived from the interface “enp1s0” and the port number.

PortID: 8c:a5:a1:ff:fe:00:00:c0, 2

syn1588® redSync

```
> redsync -A 8c:a5:a1:ff:fe:00:00:c0,1 -B 8c:a5:a1:ff:fe:00:00:c0,2 -v 3 -m simple
-e unmonitored
```

After invocation redSync checks the port states of both PTP ports and gives the control of the clock to one of PTP port; in this case to PTP port A. The following log output shows a successful invocation of redSync:

```
[ALWAYS] syn1588(R) redSync - Redundant Synchronization Engine
Build date: Oct 9 2020 - V 1.12-5 Rev gb411901
Copyright (c) Oregano Systems - Design & Consulting GesmbH 2005-2020
Confidential unpublished data - All rights reserved

[ALWAYS] Command line: /mnt/someone/bin/redsync -A 8c:a5:a1:ff:fe:00:00:c0,1 -B
8c:a5:a1:ff:fe:00:00:c0,2 -v 3 -m simple
[NOTICE] Port 0: adding config "A" = "8c:a5:a1:ff:fe:00:00:c0"
[NOTICE] Port 0: adding config "B" = "8c:a5:a1:ff:fe:00:00:c0"
[NOTICE] Port 0: adding config "v" = "3"
[NOTICE] Port 0: adding config "m" = "simple"
2020-07-16 06:17:16.294904 [INFO] Init shared mem
2020-07-16 06:17:16.295452 [INFO] Init shared mem
2020-07-16 06:17:16.295927 [ALWAYS] Redsync runs 2 times per second
2020-07-16 06:17:16.796531 [ALWAYS]
2020-07-16 06:17:16.796959 [ALWAYS] Current Iteration: 0
2020-07-16 06:17:16.797352 [ALWAYS]
2020-07-16 06:17:16.797808 [ALWAYS] [CHANGE] New Active Port A
2020-07-16 06:17:16.798198 [INFO] [RESULT] Port A : active
2020-07-16 06:17:16.798583 [INFO] [RESULT] Port B : available
2020-07-16 06:17:17.299066 [ALWAYS]
2020-07-16 06:17:17.299508 [ALWAYS] Current Iteration: 1
2020-07-16 06:17:17.299895 [ALWAYS]
2020-07-16 06:17:17.300321 [INFO] [RESULT] Port A : active
2020-07-16 06:17:17.300709 [INFO] [RESULT] Port B : available
```

```
2020-07-16 06:17:17.801191 [ALWAYS]
2020-07-16 06:17:17.801633 [ALWAYS] Current Iteration: 2
2020-07-16 06:17:17.802023 [ALWAYS]
2020-07-16 06:17:17.802460 [INFO] [RESULT] Port A : active
2020-07-16 06:17:17.802850 [INFO] [RESULT] Port B : available
2020-07-16 06:17:18.303319 [ALWAYS]
2020-07-16 06:17:18.303759 [ALWAYS] Current Iteration: 3
2020-07-16 06:17:18.304141 [ALWAYS]
2020-07-16 06:17:18.304563 [INFO] [RESULT] Port A : active
2020-07-16 06:17:18.304946 [INFO] [RESULT] Port B : available
```

If redSync has to switch the PTP port controlling the PTP hardware clock it will output these log messages:

```
2020-07-16 07:15:43.926363 [ALWAYS] Current Iteration: 124
2020-07-16 07:15:43.926777 [ALWAYS]
2020-07-16 07:15:44.427292 [ALWAYS]
2020-07-16 07:15:44.427784 [ALWAYS] Current Iteration: 125
2020-07-16 07:15:44.428206 [ALWAYS]
2020-07-16 07:15:44.428650 [ALWAYS] [CHANGE] New Active Port B
2020-07-16 07:15:44.429077 [ALWAYS] [CHANGE] Old Active Port A
2020-07-16 07:15:44.929547 [ALWAYS]
2020-07-16 07:15:44.930077 [ALWAYS] Current Iteration: 126
2020-07-16 07:15:44.930497 [ALWAYS]
2020-07-16 07:15:45.431040 [ALWAYS]
2020-07-16 07:15:45.431529 [ALWAYS] Current Iteration: 127
2020-07-16 07:15:45.431923 [ALWAYS]
```

Example – Scenario B

The setup of the system is as depicted in Figure 2.

Two syn1588[®] PTP Stack instances have to be invoked in the background and have to synchronize to two different PTP grandmaster instances over the network.

Two syn1588[®] PCIe NIC are required for this setup.

In addition to redSync, an instance of the syn1588[®] ISync utility has to be invoked.

For the sake of demonstration, Port B will be synchronizing to a free-running PTP Grandmaster that provides a PTP time that is offset at the time of this test to a few hundred ns to the PTP Grandmaster on Port A, which is synchronized to GPS.

syn1588[®] PCIe NIC Port A

For this example a syn1588[®] PCIe NIC is mounted in the PC and named “enp1s0”.

The MAC address of this NIC is: 00:1e:c0:f6:c7:3d and the clockID for PTP operation is derived from it.

clockID: 00:1e:c0:ff:fe:f6:c7:3d

syn1588[®] PCIe NIC Port B

For this example a syn1588 PCIe NIC is mounted in the PC and named “enp2s0”.

The MAC address of this NIC is: 00:1e:c0:85:d0:0c and the clockID for PTP operation is derived from it.

clockID: 00:1e:c0:ff:fe:85:d0:0c

syn1588[®] PTP Stack for Port A

```
> ptp -I enp1s0 -d 51
```

... to synchronize to a PTP Grandmaster (PTP domain: 51) over the local Ethernet interface “enp1s0”

This PTP instance can be identified by its clockID (derived from the interface “enp1s0” and the port number (default: 1).

PortID: 00:1e:c0:ff:fe:f6:c7:3d, 1

syn1588® PTP Stack for Port B

```
> ptp -I enp2s0 -d 50
```

... to synchronize to a PTP Grandmaster (PTP domain: 50) over the local Ethernet interface “enp1s0”

This PTP instance can be identified by its clockID (derived from the interface “enp2s0” and the port number (default: 1).

PortID: 00:1e:c0:ff:fe:85:d0:0c, 1

syn1588® redSync

```
> ./redSync -A 00:1e:c0:ff:fe:f6:c7:3d,1 -B 00:1e:c0:ff:fe:85:d0:0c,1 -m bmc -v 3
-l -e unmonitored
```

After invocation redSync checks the port states of both PTP ports and gives the control of the clock to one of the PTP ports; in the example shown below to PTP port A. The following log output shows a successful invocation of redSync:

```
[ALWAYS] syn1588(R) redSync - Redundant Synchronization Engine
Build date: based on v1.13-2
Copyright (c) Oregano Systems - Design & Consulting GesmbH 2005-2021
Confidential unpublished data - All rights reserved

[ALWAYS] Command line: ./redSync -A 00:1e:c0:ff:fe:f6:c7:3d,1 -B
00:1e:c0:ff:fe:85:d0:0c,1 -m bmc -v 3 -l -e unmonitored
[NOTICE] Port 0: adding config "A" = "00:1e:c0:ff:fe:f6:c7:3d"
[NOTICE] Port 0: adding config "B" = "00:1e:c0:ff:fe:85:d0:0c"
[NOTICE] Port 0: adding config "m" = "bmc"
[NOTICE] Port 0: adding config "v" = "3"
[NOTICE] Port 0: adding config "l" = ""
[NOTICE] Port 0: adding config "e" = "unmonitored"
[WARNING] [WARNING] More then one clock id was given!
2021-07-21 11:30:59.686628 [INFO] Init shared mem
2021-07-21 11:30:59.686728 [INFO] Init shared mem
2021-07-21 11:30:59.686798 [ALWAYS]
2021-07-21 11:30:59.686815 [ALWAYS] Current Iteration: 0
2021-07-21 11:30:59.686832 [ALWAYS]
2021-07-21 11:30:59.686923 [ALWAYS] [Port A] is available but not active!
2021-07-21 11:30:59.687008 [ALWAYS] [Port B] is available but not active!
2021-07-21 11:30:59.687027 [ALWAYS] [CHANGE] Detected! At Port A
2021-07-21 11:30:59.687057 [ALWAYS] Port A sets system clock adjustment to 1
2021-07-21 11:30:59.687075 [ALWAYS] [CHANGE] New Active Port A
2021-07-21 11:30:59.687093 [INFO] [RESULT] Port A : active
2021-07-21 11:30:59.687110 [INFO] [RESULT] Port B : available
2021-07-21 11:31:00.687242 [ALWAYS]
2021-07-21 11:31:00.687280 [ALWAYS] Current Iteration: 1
```

The following log snippet displays the behavior of redSync when it detects an issue on the currently selected port. The issue is resolved by switching to the other port and, in this scenario, trigger a switch to Port B also for the connected utility ISync.

```

2021-07-21 11:33:45.745560 [ALWAYS] [Port A] is active!
2021-07-21 11:33:45.745603 [ALWAYS] [Port B] is available but not active!
2021-07-21 11:33:45.745630 [INFO] [RESULT] Port A : active
2021-07-21 11:33:45.745655 [INFO] [RESULT] Port B : available
2021-07-21 11:33:46.745792 [ALWAYS]
2021-07-21 11:33:46.745831 [ALWAYS] Current Iteration: 167
2021-07-21 11:33:46.745851 [ALWAYS]
2021-07-21 11:33:46.745899 [INFO] [Port A] No new data can be fetched from
ptp. Check if it is still running !
2021-07-21 11:33:46.745921 [WARNING] [Port A] Port is not available !
2021-07-21 11:33:46.745971 [ALWAYS] [Port B] is available but not active!
2021-07-21 11:33:46.745996 [INFO] [RESULT] Port A : not available
2021-07-21 11:33:46.746020 [ALWAYS] [CHANGE] Detected! At Port B
2021-07-21 11:33:46.746069 [ALWAYS] Port B sets system clock adjustment to 1
2021-07-21 11:33:46.746094 [ALWAYS] [CHANGE] New Active Port B
2021-07-21 11:33:46.746116 [ALWAYS] [CHANGE] Old Active Port A
2021-07-21 11:33:46.746139 [INFO] [RESULT] Port B : active
2021-07-21 11:33:47.746278 [ALWAYS]
2021-07-21 11:33:47.746317 [ALWAYS] Current Iteration: 168
2021-07-21 11:33:47.746338 [ALWAYS]
2021-07-21 11:33:47.746388 [INFO] [Port A] No new data can be fetched from
ptp. Check if it is still running !
2021-07-21 11:33:47.746415 [WARNING] [Port A] Port is not available !
2021-07-21 11:33:47.746465 [ALWAYS] [Port B] is active!
2021-07-21 11:33:47.746493 [INFO] [RESULT] Port A : not available
2021-07-21 11:33:47.746512 [INFO] [RESULT] Port B : active
2021-07-21 11:33:48.746651 [ALWAYS]
2021-07-21 11:33:48.746690 [ALWAYS] Current Iteration: 169

```

syn1588® ISync

The ISync utility is used to synchronize the system clock to the PTP controlled hardware clock. Together with redSync, ISync chooses the better PTP Port, hence, the more accurate PTP Grandmaster. With the support of redSync, ISync switches automatically to the other PTP Grandmaster, if the connection to the original Grandmaster is lost or degrades in performance.

```
> ./lSync -v 2 -r -b 100ns
```

The parameter “-b 100ns” configures ISync to switch to the alternate Port selected by redSync, only if it is within a certain boundary, in this case 100 ns.

The following snippet shows the first few seconds of operation.

```
syn1588(R) lSync - Local Synchronization engine
Build date: 2021-07-19T12:12:58 - v1.13-2-ge2191f37
Copyright (c) Oregano Systems - Design & Consulting GesmbH 2005-2021
Confidential unpublished data - All rights reserved

Command line: ./lSync -v 2 -r -b 100ns
Port 0: adding config "v" = "2"
Port 0: adding config "r" = ""
Port 0: adding config "b" = "100ns"
Syn1588Ifc requires at least:
- linux driver version 1.4-15-g05b7283
- windows driver version 10/05/2017, 10.9.16.182
Syn1588Impl: Device /dev/syncD0 found
syn1588(R) Hardware Clock M 2.3.5 f=125000000 Hz
syn1588(R) PCIe NIC Revision 2, Build 865 with HQ Oscillator
Syn1588Impl: Device /dev/syncD1 found
syn1588(R) Hardware Clock M 2.3.5 f=125000000 Hz
syn1588(R) PCIe NIC Revision 2, Build 865 with HQ Oscillator
Found 3 clocks
Clk: Using Oregano Systems; syn1588(R) PCIe NIC Revision 2; 00:1E:C0:F6:C7:3D
    with ClockId 00:00:00:00:00:00:00
Clk: Resetting servos
Clk: Resetting filters
Clk: Using Oregano Systems; syn1588(R) PCIe NIC Revision 2; 00:1E:C0:85:D0:0C
    with ClockId 00:00:00:00:00:00:00
Clk: Resetting servos
Clk: Resetting filters
Clk: Using Linux System Clock;;
    with ClockId 00:00:00:00:00:00:00
Clk: Resetting servos
Clk: Resetting filters
Syn1588Impl: readReg error: 0x0200
Syn1588Impl: writeReg error: 0x0200 - 0x00000009
Syn1588Impl: readReg error: 0x0200
Syn1588Impl: writeReg error: 0x0200 - 0x00900000
High precision sync activated for redSync operation.
PathDelay 12 ns
PathDelay 12 ns
Starting SyncEngine with 3 clocks
Syn1588Impl: readReg error: 0x1600
Syn1588Impl: readReg error: 0x1604
Syn1588Impl: readReg error: 0x1608
Syn1588Impl: readReg error: 0x160c
Changing Synchronization Origin to syn1588 0
Wait for Clock Event timed out
Updated UTCOffset to SystemClock
syn1588 0: 1626867111.434708005 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867111.435250863
current UTC offset: 37 sec

not adjusting clock syn1588 1 in slave state..
```

```

Adjusting Local:
Update M2S-Delay 542891 ns
Waiting for calibration...

syn1588 0: 1626867112.000133775 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867112.000676833
current UTC offset: 37 sec

not adjusting clock syn1588 1 in slave state..
Adjusting Local:
Update M2S-Delay 543091 ns

syn1588 0: 1626867113.000131835 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867113.000675185
current UTC offset: 37 sec

not adjusting clock syn1588 1 in slave state..
Adjusting Local:
Update M2S-Delay 543379 ns

syn1588 0: 1626867114.000131615 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867114.000675260
current UTC offset: 37 sec

not adjusting clock syn1588 1 in slave state..
Adjusting Local:
Update M2S-Delay 543674 ns

syn1588 0: 1626867115.000106579 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867115.000650532
current UTC offset: 37 sec

not adjusting clock syn1588 1 in slave state..
Adjusting Local:
Update M2S-Delay 543982 ns

syn1588 0: 1626867116.000136983 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867116.000681212
current UTC offset: 37 sec

not adjusting clock syn1588 1 in slave state..
Adjusting Local:
Update M2S-Delay 544256 ns
Adjusting clock at 544256.00 ns offset

```

ISync detects two [®]syn1588 hardware units, “syn1588 0” and “syn1588 1”, it uses “syn1588 0” = “Port A” as clock source, which has been qualified by redSync.

As can be seen from the last line, after a few seconds ISync starts to synchronize the system clock to the “Origin”.

At this point the system clock is offset by 544 μ s to the source clock.

The following snippet shows the behavior of ISync during the switch from Port A to Port B as initiated by redSync at 2021-07-21 11:33:46.

```

syn1588 0: 1626867260.000102277 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867260.000102208
current UTC offset: 37 sec

not adjusting clock syn1588 1 in slave state..
Adjusting Local:
Update M2S-Delay -40 ns
Adjusting clock at -53.64 ns offset

Warning: Found no running PTP Stack for syn1588 0
syn1588 0: 1626867261.000100608 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867261.000100557
current UTC offset: 37 sec

Waiting for origin to be synchronized within boundary and being selected...
Warning: Found no running PTP Stack for syn1588 0
syn1588 0: 1626867262.000104572 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867262.000104519
current UTC offset: 37 sec

Waiting for origin to be synchronized within boundary and being selected...
Warning: Found no running PTP Stack for syn1588 0
syn1588 0: 1626867263.000100936 (Origin)
syn1588 1: 1626699692.000000018
Local:    1626867263.000100896
current UTC offset: 37 sec

Waiting for origin to be synchronized within boundary and being selected...
Warning: Found no running PTP Stack for syn1588 0
Changing Synchronization Origin to syn1588 1
syn1588 0: 1626867264.000000020
syn1588 1: 1626699692.000000018 (Origin)
Local:    1626867264.000100850
current UTC offset: 37 sec

not adjusting clock syn1588 0 in slave state..
Adjusting Local:
Update M2S-Delay 0 ns
Clk: Timestamp out of range: 1626699692.000000018

Warning: Found no running PTP Stack for syn1588 0
syn1588 0: 1626867264.000000020
syn1588 1: 1626867265.000102690 (Origin)
Local:    1626867265.000102634
current UTC offset: 37 sec

not adjusting clock syn1588 0 in slave state..
Adjusting Local:
Update M2S-Delay -30 ns
Adjusting clock at -42.82 ns offset

```

ISync has been synchronizing the system clock up to this point to Port A (syn1588 0). The connection to the PTP Grandmaster on Port A is capped, detected by redSync and communicated to ISync. Which selects Port B as alternative and executes a seamless transition from PTP Grandmaster connected via Port A to the PTP grandmaster connected via Port B.

As can be seen from the above log snippet, the synchronization accuracy stays within +/- 50 ns during the switch from Port A to Port B.

 <p>Oregano Systems A Meinberg Company</p> <p>Franzosengraben 8 A-1030 Vienna Austria http://oregano.at contact@oregano.at</p>	<p>Copyright © 2021 Oregano Systems – Design & Consulting GmbH ALL RIGHTS RESERVED.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>All trademarks used in this document are the property of their respective owners.</p>
--	--