

Abstract

redSync is another utility of the syn1588® software suite. redSync allows the redundant synchronization of a PTP slave. redSync is based on the syn1588® PTP Stack as well as syn1588® hardware. The redSync utility is available starting with the release Q3/2020 of the syn1588® software suite. This application note describes the basic usage of this utility.

Introduction

The basic application structure for redSync is as follows. Two independent instances of the syn1588® PTP Stack connect to one PTP Grandmaster each.

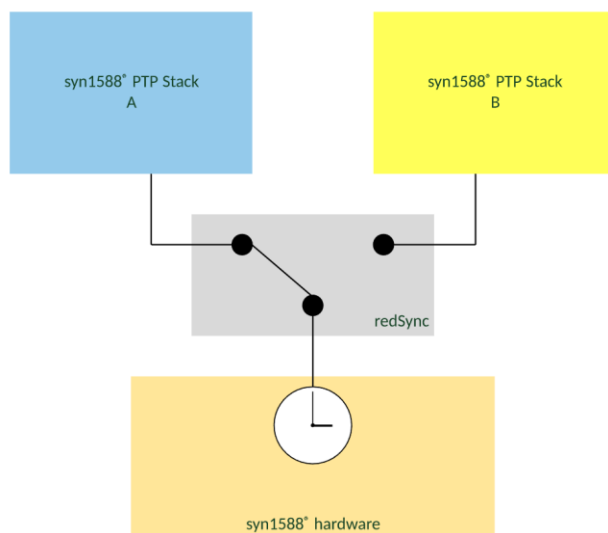


Figure 1: redSync: basic setup

redSync observes the synchronization status of these two syn1588® PTP Stack instances and selects on a configurable criteria the syn1588® PTP Stack instance that actually shall maintain and control the accurate high-precision PTP hardware clock. If e.g. 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.

Current Limitations

Note, redSync is currently just supported on Linux OS. Windows support will be added with the next releases.

Note, redSync currently supports a single syn1588[®] PCIe NIC for the hardware interface. Support for other syn1588[®] hardware configurations will be added with the next releases.

Note, just IPv4 support had been tested right now for this release. IPv6 support is implemented but not tested. Layer 2 support is currently not implemented.

Pre-requisites

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 runs Linux
- 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

The currently 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.

Operation

redSync connects to the two syn1588[®] PTP Stack instances specified on the command line via their respective Shared Memory interface. Depending on the selected decision algorithm redSync decides which of the two syn1588[®] PTP Stack shall maintain the syn1588[®] hardware clock and thus actually synchronize the node's clock. This selection is done by setting "noClockAdjustment" parameter via the Shared Memory interface. If redSync has to decide to switch from one syn1588[®] PTP Stack instance to another it will keep the PTP slave clock within an acceptable range with respect to the Grandmaster.

redSync will start its operation only after if there are two valid connections to syn1588[®] PTP Stack instances which both are in PTP slave state. Currently, there are two operation modes, which defines the decision criteria between the two syn1588[®] PTP Stack instances.

Note, it is also possible to use the redSync with two syn1588[®] PTP Stack instances both running in software mode, i.e. without syn1588[®] hardware. In this use case, the selected PTP port will adjust the system clock.

Even if the redSync takes care of the organization of the clock adjusting between both PTP ports, the following PTP configuration is recommended for all use cases in order to avoid the changes at the clock prior a decision is met:

- Both ports shall be instructed not to allow clock adjustments

This PTP configuration is done via the "noClockAdjustment" parameter

Overview of Modes

redSync supports several modes of operation. There are environment modes that reacts on the state of the environment and decision modes which can be separately selected:

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

Monitored mode

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 need 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 are running for both PTP ports.
- Both PTP ports are in slave state.

Unmonitored mode

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.

Simple Mode (default mode)

In this mode redSync is instructed to choose the first given PTP port A as default. redSync will allow this PTP port A to actually control and maintain the hardware clock. As long as the PTP port A owns a connection to a PTP Grandmaster which can be used for synchronization the PTP port A will continue to control the hardware clock.

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 owns again a connection to a PTP Grandmaster which can be used for synchronization. redSync will switch back in such an event and PTP port A is again allowed to the control the hardware clock. PTP port B actually acts as a backup connection for PTP port A in this mode of operation.

BMC Mode

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 may origin 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 control and maintain the hardware clock. If that active PTP port loses the connection to its Grandmaster or the Grandmaster's performance is degraded than the other PTP port will take the control of the hardware clock.

Command Line Parameters

These are the command line parameters 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: ./redsync -h
[ALWAYS]
[ALWAYS] -----
[ALWAYS] Usage : redsync -A <clock_id,port_nr>
[ALWAYS]         -B <clock_id,port_nr>
[ALWAYS]         | [-h] [-v <loglevel>] [-m <compare method>] [-e <environment
state>]
[ALWAYS]
[ALWAYS] -h             shows this usage screen
[ALWAYS] -v <loglevel> change verbosity level (0..4) [2]
[ALWAYS] -m <compare method> method to compare the masters [simple(default) , bmc ]
[ALWAYS] -e <environment state> [unmonitored(default) , monitored ]
[ALWAYS] -----
[ALWAYS] -A and -B are mandatory and set the two PTP instances
[ALWAYS] managed by the redsync utility, they are defined by the
[ALWAYS] following two fields:
[ALWAYS]
[ALWAYS] <clock_id, : clock id, 16 digits in hexadecimal notation
[ALWAYS] for example : 8ca5a1ffffe000925
```

```

[ALWAYS]         port_nr> : port number, integer between 1 - 65535,
[ALWAYS]         typical value: 1 or 2 for common setups
[ALWAYS] -----
[ALWAYS] Example : redsync -A 8c:a5:a1:ff:fe:00:02:05,1 -B 8c:a5:a1:ff:fe:00:02:05,2 -v 3 -m
[ALWAYS] simple
[ALWAYS]         or : redsync -A 8ca5alfffe000205,1 -B 8ca5alfffe000205,2 -v 3 -m bmc
[ALWAYS] -----
[ALWAYS] Extended help:
[ALWAYS]
[ALWAYS] How can i find clock id(s) and/or port number(s)?:
[ALWAYS]
[ALWAYS] OPTION 1 - Normal PTP stack execution:
[ALWAYS] Usage: > ./ptp <configuration file> || <command line parameters>
[ALWAYS] IMPORTANT: Do not forget to set the log verbosity level to 3 or 4!
[ALWAYS] configuration file parameter: "logLevel 3" or "logLevel 4"
[ALWAYS] command line parameter: "-v 3" or "-v 4"
[ALWAYS] search for a line matching the following:
[ALWAYS] "(x) with ClockId YY:YY:YY:YY:YY:YY:YY:YY"
[ALWAYS] x ... <port_nr>
[ALWAYS] YY:YY ... <clockId>
[ALWAYS]
[ALWAYS] OPTION 2:
[ALWAYS] <clock_id> :
[ALWAYS] You can use the syn1588 utility to query the clock id.
[ALWAYS] Usage : > ./syn1588
[ALWAYS] >>listcard //list the clock ids of NICs on the system.
[ALWAYS] e.g: >>card 0 8c:a5:a1:ff:fe:00:09:25
[ALWAYS]
[ALWAYS] <port_nr> :
[ALWAYS] - PTP Ports are defined in the configuration file for the PTPv2 stack.
[ALWAYS] Each port is named as follows: "port<x>:", where <port_nr> = x
[ALWAYS] - If you are starting the PTP stack with command line parameters
[ALWAYS] (e.g., -P to create multiple ports),
[ALWAYS] + On the same network interface = same <clock_id>:
[ALWAYS] the first port will have port_nr = 1 and
[ALWAYS] further ports will have 2, 3, 4, ...
[ALWAYS] + On different network interfaces = different <clock_id>:
[ALWAYS] for common setups both <port_nr> = 1
[ALWAYS]
[ALWAYS] @-v <logLevel>:
[ALWAYS] 0 ... [ALWAYS]
[ALWAYS] 1 ... [ALWAYS], [ERROR]
[ALWAYS] 2 ... [ALWAYS], [ERROR], [WARNING]
[ALWAYS] 3 ... [ALWAYS], [ERROR], [WARNING], [INFO]
[ALWAYS] 4 ... [ALWAYS], [ERROR], [WARNING], [INFO], [DEBUG]
[ALWAYS]
[ALWAYS] @-m <compare method>:
[ALWAYS] simple: port A is the active master as long as it is available.
[ALWAYS] bmc : BestMasterClock algorithm is used to decide the master
[ALWAYS]
[ALWAYS] @-e <environment state>:
[ALWAYS] monitored : PTPv2 stack instances are being monitored and checked if they
[ALWAYS] are ready for redSync.
[ALWAYS]
[ALWAYS] redSync expects all PTP ports to be synchronized to a master
[ALWAYS] when operation starts.
[ALWAYS]
[ALWAYS] If this condition is not satisfied, it terminates.
[ALWAYS] It will also terminate if, during operation,
[ALWAYS] all PTP ports loose synchronization with their respective
[ALWAYS] Master.
[ALWAYS]
[ALWAYS] unmonitored : redSync will start and will keep running, even when there are
[ALWAYS] no PTPv2 stacks operational.

```

Example

A typical application example for the redSync is as follows. One PC - or in general a PTP node - with one physical network interface is connected to a network. The PTP node is equipped with a syn1588® PCIe NIC. Two syn1588® PTP Stack instances run on the PC using the syn1588® PCIe NIC. Both syn1588® PTP Stack instances run in PTP slave mode (either controlled by invocation or by BMCA). Both syn1588® PTP Stack instances already established a valid connection to their PTP Grandmasters on the network.

redSync is started as shown in the following command example.

```
> 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 the example shown above 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]

```

The following log output shows the error messages displayed by redSync started in monitored mode if the startup tests (both PTP ports active and in PTP slave mode) fail:

```

[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 -e monitored
[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 "e" = "monitored"
2020-07-16 07:39:51.833471 [INFO] Init shared mem
2020-07-16 07:39:51.833546 [INFO] Init shared mem
2020-07-16 07:39:51.833599 [ALWAYS] redsync runs 2 times per second
2020-07-16 07:39:54.834263 [ERROR] [ERROR] RedSync can not be started.
2020-07-16 07:39:54.834291 [ERROR] [CHECK] All syn1588@ PTP Stack's shall
run and all ports shall be in PTP slave mode!
2020-07-16 07:39:54.834304 [NOTICE] Closing shared mem 11785127
2020-07-16 07:39:54.834371 [NOTICE] Closing shared mem 11730957
2020-07-16 07:39:54.834400 [NOTICE] Closing shared mem 11829265
2020-07-16 07:39:54.834427 [NOTICE] Closing shared mem 11796496

```




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

Copyright © 2020

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.