Solarflare Enhanced PTP User Guide

SF-109110-CD

Issue 2

Table of Contents

Chapter 1: What's New

Overview

This document is the user guide for **Solarflare Enhanced PTP** (sfptpd) which is an enhanced PTP daemon for use with Solarflare adapters supporting:

- Hardware timestamps offered by the SFN5322F, SFN6322F, SFN7322F (and any other SFN7000 series adapter with appropriate AppFlex™ license).
- The ability to synchronize the high precision clock on multiple adapters - one of the adapter clocks is treated as the "Local Reference Clock" and is used to synchronize the server's system clock and clocks on other adapters.
- PTP hybrid mode which allows a mix of multicast and UDP unicast transmission methods for PTP messages between PTP master and slave clocks.
- PTP over VLAN interfaces.
- PTP over active/standby bonded interfaces.

This issue of the user guide supports sfptpd from version 2.1.0.33 which includes full support for the Solarflare Flareon™ Ultra SFN7122F and SFN7322F adapters.

New Features

NTP Synchronization Modes

sfptpd v2.1.0.33 supports two NTP synchronization modes:

- Configured as a PTP master clock, the server running sfptpd can also run an NTP client synchronizing to an NTP server. The master transmits PTP packets to all slaves on the PTP network.
- The SFN7000 series adapters can generate hardware timestamps for all network packets received on an interface. Even when PTP is not being used, sfptpd can still be used to synchronize the clock on the adapter(s) to the system's view of time and hardware timestamps can be compared with system time. When sfptpd is used in this mode, NTP can be used in parallel to synchronize the system clock to an upstream clock reference. sfptpd is then solely used to discipline all adapter clocks in the local server using the system clock as a reference.

Timestamping Interfaces

On Solarflare 7000 series adapters, sfptpd can be used to enable hardware timestamping of all packets (to the Linux kernel) on specified interfaces. For details of hardware timestamping configuration see Hardware Timestamps on page 26.

Improved Alarm and Status Reporting

The sfptpd statistics output is now colour coded to identify alarm conditions. Values will appear in red text when viewed on a standard terminal device to indicate problems in the PTP message sequence.

The PTP network topology file now includes additional status reporting, an improved topology clock hierarchy map and PTP network alarm states including alarms to detect loss of Sync, Follow_up and Delay_Resp messages. See The Toplogy File on page 34 for an example and description of how to use the topology file.

New Parameters

The configuration file supports the following new options

Table 1: New Configuration File Options

Option	Description
daemon	Run sfptpd as a daemon. Disabled by default.
timestamping-interfaces	Solarflare Flareon™ SFN7000 series adapters only. Identify the set of interfaces on to enable hardware timestamping of all packets (to the Linux kernel). timestamping-interfaces <name | MAC address | *> e.g `timestamping-interfaces eth2 eth5 eth9`
timestamping-disable-on-exit	Solarflare Flareon™ SFN7000 series adapters only. Specify whether timestamping of all packets (to the Linux kernel) should be disabled when the sfptpd process or daemon exits. `timestamping-disable-on-exit <off | on>`
freerun-mode	Used together with the 'sync-mode freerun' setting. This option identifies whether to synchronize all clocks to an adapter clock or synchronize all clocks to the system clock while allowing NTP to run. `freerun-mode <nic | ntp>`

Documentation Changes

- For users new to PTP, Chapter 3 provides a basic introduction to PTP messages, the synchronization process and describes the output generated by Solarflare sfptpd.
- The improved PTP network topology file is explained in The Toplogy File on page 34.

Chapter 2: Introduction

2.1 Purpose

This document describes Solarflare Enhanced PTP (sfptpd) support for Solarflare's 10GbE SFP+ Time Synchronization Server Adapters. These adapters support hardware time stamps of PTP packets and can be deployed in networks where there is a requirement to support the IEEE 1588 Precision Time Protocol. Adapters supported by sfptpd:

- Solarflare SFN5322F Dual-Port 10GbE Precision Time Stamping Server Adapter
- Solarflare SFN6322F Dual-Port 10GbE SFP+ Server Adapter
- Solarflare SFA6902F Dual-Port 10GbE SFP+ ApplicationOnload™ Engine
- Solarflare Flareon™ Ultra SFN7322F Dual-Port 10GbE PCIe 3.0 Server I/O Adapter
- Any Solarflare Flareon™ SFN7000 series adapter with a PTP/timestamping AppFlex™ license installed

The document describes installation and configuration procedures for the network adapter and software components needed to run PTP on Solarflare adapters.

2.2 Definitions, Acronyms and Abbreviations

1PPS	1 Pulse Per Second
LRC	Local Reference Clock - the active clock to which all other clocks, on a PTP enabled server, are synchronized
NTP	Network Time Protocol
PID	Proportional, Integral, Derivative filter
PPB	Parts Per Billion
PPM	Parts Per Million
PTP	Precision Time Protocol
ptpd2	Original Solarflare PTP daemon - implementation of IEEE-1588-2008 (PTP version 2)
sfptpd	Solarflare Enhanced PTP daemon - implementation of IEEE-1588-2008 (PTP version 2)
UUID	Universally Unique Identifier
VLAN	Virtual Local Area Network

2.3 Transition Guidelines

Solarflare recommend that users of Solarflare (legacy) ptpd2 transition to sfptpd to take advantage of the advanced features, not available to ptpd2, and to benefit from future Solarflare PTP development.

Solarflare Enhanced PTP employs a configuration file in place of the traditional command line options to simplify the startup and configuration process. For comparison of command line options with configuration file variables refer to Configuration Files on page 31 and Appendix C: Transition Guide on page 48.

NOTE: Users of the previous Solarflare PTP implementation ptpd2 should refer to the Solarflare (Legacy) PTP User Guide (SF-107094-CD). This document refers only to Solarflare Enhanced PTP - sfptpd.

Chapter 3: How PTP Works

This section provides a basic description of how the PTP protocol operates between a master and a slave server. For a complete description of the PTP protocol refer to the IEEE 1588-2008 Standard for a Precision Clock.

3.1 Message Sequence

The following diagram describes the PTP protocol message sequence which must occur for master and slave servers to synchronize.

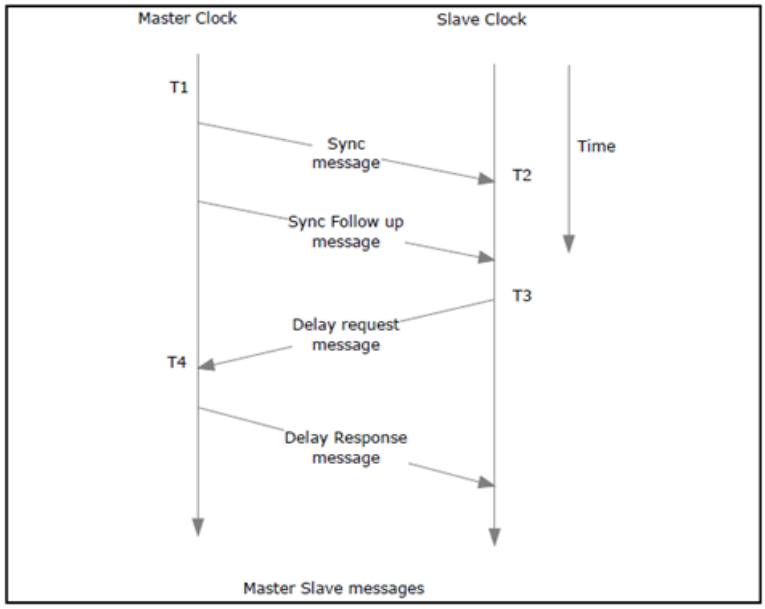

Figure 1: PTP Message Sequence

The **Sync message** is multicast to all slaves at a fixed interval of between 1 and 64 messages per second, configurable by the master clock. On most PTP networks a sync interval of between 1-4 sync messages per second is sufficient to ensure accurate synchronization and increasing the sync interval does not always result in greater accuracy of synchronization. The Sync message contains the time the message was transmitted (T1). The slave generates a hardware timestamp (T2) when the message is received.

The **Follow_up message** is sent immediately following every Sync by master clocks using 2-step synchronization. The Follow_up message contains the actual time the preceding Sync message was sent. A master clock using 1-step synchronization does not transmit the Follow_up message.

When the slave has received the Follow_up message (or just Sync message in the case of 1-step synchronization) it will generate a **Delay_Request message**. When this message is sent the slave generates and retains a hardware timestamp (T3).

The master will record the time the Delay_Request is received (T4) and this timestamp is then relayed back to the slave in the **Delay_Response message**.

Using the timestamp information derived from the message sequence, the slave is able to calculate the one-way-delay between slave and master clocks and the time offset from the master clock.

```
one-way-delay=((T2-T1) + (T4-T3)) / 2     offset=((T2-T1) - (T4-T3)) / 2
```

3.2 One-Way-Delay Interval

The interval between Delay_Request messages is determined by the master clock. A parameter of the Delay_Response message from the master is the logMessagePeriod. This is a power of 2 value that defines a send window period designed to ensure (1) that multiple slaves send at random intervals during the period, (2) that the master clock is able to respond to Delay_Requests from multiple slaves without queing these messages.

```
window = (2^logMessagePeriod)*2
```

So a logMessagePeriod of 3:

```
window = (2^3)*2 = 0-16 second window
```

Solarflare sfptpd will allow the slave to override the logMessagePeriod using the config file option `ptp-delayreq-interval` causing the sfptpd slave to send Delay_Request messages at a fixed interval.

3.3 Announce Message

Another message periodically generated by the master clock is the **Announce message** which contains data describing the master clock type, accuracy and priority levels. The Announce message is used by the Best Master Clock algorithm to determine the most accurate master clock on a PTP network.

3.4 Solarflare sfptpd 2-stage synchronization

The PTP messages are used by sfptpd to synchronize the adapter clock with the master clock. sfptpd runs a second clock servo to synchronize the system clock to the adapter clock as illustrated by Figure 2. This unique two stage synchronization has a number of benefits including:

- Improved accuracy with the ability to more frequently discipline the system clock than is supported by standard PTP masters.
- Ability to discipline the system clock to a high precision clock during periods, for whatever reason, whereby the upstream PTP master is inaccessible or offline.

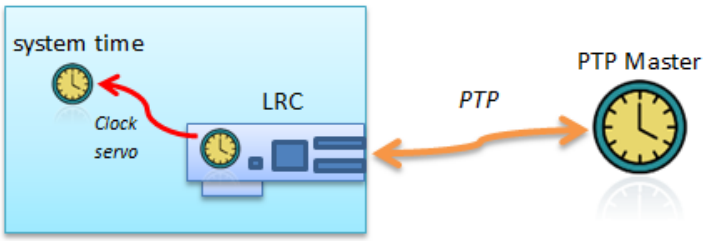

Figure 2: sfptpd 2-stage synchronization

3.5 Understanding the sfptpd startup sequence

When the sfptpd daemon is started it will generate several lines of output. A typical startup sequence, as the slave graduates from an initial **start** to a **listening** state and finally to a **slave** state, is shown below (line numbers added) with descriptions in the following table.

```
[slave-server]# ./sfptpd -i eth2 -fconfig/ptp_slave.cfg

1 info: Solarflare Enhanced PTP Daemon, version 2.1.0.32
2 info: no clock frequency correction file /var/lib/sfptpd/freq-correction-
     000f:53ff:fe21:9bb0
3 info: no clock frequency correction file /var/lib/sfptpd/freq-correction-system
4 info: using PTP sync-module
5 info: PTP clock: local reference clock is phc0(eth2/eth3), PTP clock is phc0(eth2/
     eth3)
6 info: interface eth2: SO_TIMESTAMPING enabled
7 notice: trying SO_TIMESTAMPING...
8 notice: SO_TIMESTAMPING enabled
9 info: Refreshed IGMP multicast memberships
10 info: === Now in state: PTP_LISTENING
11 info: ptp module is 0xa914e0
12 [phc0(eth2/eth3)->system], offset: 6627588.125, freq-adj: -2859245.408, in-sync:
     0
13 [phc0(eth2/eth3)->system], offset: 3777396.000, freq-adj: -1876237.065, in-sync:
     0
14 [phc0(eth2/eth3)->system], offset: 1781459.000, freq-adj: -1160973.804, in-sync:
     0
15 [phc0(eth2/eth3)->system], offset: 505077.812, freq-adj: -684452.859, in-sync: 0
```

```
16 info: === Now in state: PTP_SLAVE, Best master: 000f53fffe160474(unknown)/01
```

Table 2: sfptpd startup output

Line #	Description
1	Version of sfptpd running.
2-3	A frequency correction file holds the frequency correction value (PPB) that is currently being used to discipline a clock. On initial startup frequency correction files do not exist. Once created by sfptpd for each clock, they are updated every 60 seconds and they are preserved over server reboot and sfptpd restart.
4	The synchronization mode being used - set with the config file `sync-mode` option.
5	The Local Reference Clock. On a 5000/6000 series adapter this will identify the interface of the adapter clock. On a 7000 series adapter this identifies the PTP hardware clock in the form: phc0(ethX/ethY) where ethX is the active clock interface and ethY the second adapter clock interface on this adapter. Both interfaces on a 7000 series adapter share the same clock.
6-8	On older kernels, from 2.6.18, that predate SO_TIMESTAMPING, an IOCTL interface is used for time stamping PTP packets. If hardware timestamps cannot be initialized sfptpd will revert to software timestamping using the system clock.
9	sfptpd joins the multicast group for PTP traffic using address 224.0.1.129.
10	The slave remains in a LISTENING state listening for Announce messages from any master clock on the network. The Best Master Clock algorithm is used to determine the most accurate master clock to which all slaves will synchronize. Any other master clock on the same network will go into passive mode and not send PTP messages once the best master has been selected. The Best Master Clock algorithm is implemented in such a way that all slaves will arrive at the same conclusion and select the same master clock.
11	The sfptpd module identifier.
12-15	When sfptpd is running it will immediately begin disciplining the system clock against the adapter clock using the adapter's precision oscillator. This will happen even if PTP messages are not being received. Before the adapter clock can begin synchronization with the upstream master clock, the slave must receive an Announce message followed by Sync and Follow_up messages.
16	The slave moves to a SLAVE state once it has received an Announce message, and selected the best master clock - identified by UUID derived from the MAC address. Following this, the slave will accept PTP Sync messages from the master and start to synchronize the adapter clock(s) with the master clock.

3.6 Understanding sfptpd output

Once it has reached a SLAVE state, sfptpd will continue to generate output describing the current state of the synchronization.

The following examples are from a slave server

```
[ptp-gm->phc0(eth2)], offset: 235.000, freq-adj: -1444.729, in-sync: 1, one-way-
     delay: 932.000
```

Table 3: sfptpd offset output

Parameter	Description
`[ptp-gm->phc0(eth2)]`	The values on this line show offset and one-way-delay between the master clock and the local identified clock.
`freq-adj`	The current rate (PPB) at which the clock is being disciplined by sfptpd. This value is stored in the freq-correction file for this clock every 60 seconds.
`offset`	The current offset (nanoseconds) between the adapter clock and the master clock. Immediately following startup this is expected to be a large value, but will gradually decrease until it settles to its lowest value. Synchronization can typically take between 15-30 minutes. From sfptpd version 2.1.0.32 the offset value will be shown as RED text if an alarm condition exists which affects the synchronization - Check the topology file for current alarms status.
`in-sync`	The in-sync flag will be 1 when the offset between master and slave clocks is below 1 microsecond for a period of 1 minute. The in-sync flag will be 0 before the above condition is true. Using sfptpd 2.1.0.33 and later, the in-sync flag will change to 0 if an alarm condition exists on the server to indicate problems in the PTP network e.g. PTP messages not being sent or received by the slave server. Check the topology file for current alarms status.

Table 3: sfptpd offset output

Parameter	Description
`one-way-delay`	The current one-way-delay (nanoseconds) between master and slave servers. This value should not be zero, but, once the server is synchronized, it should remain fairly stable. If the value is zero - check that Delay_Req and Delay_Resp message are being sent and received. If the value does not change at all over an extended period - check the Delay_Req interval. From sfptpd version 2.1.0.32 the offset value will be shown as RED text if an alarm condition exists which affects the synchronization. Check the topology file for current alarms status.

By default sfptpd will output the offset data to stdout. It is possible to log this to file using the configuration file `stats-log` parameter.

Even when logging to file it is possible to override this using the -v option on the sfptpd command line.

Chapter 4: Overview

4.1 Solarflare PTP Network Adapters

Solarflare time synchronization adapters are dual port SFP+ 10G Ethernet network server adapters that can generate hardware timestamps for PTP packets in support of a network precision time protocol deployment, and in accordance with the IEEE 1588-2008 specifications. With hardware precision and performance, the PTP adapters facilitate PTP slave servers to accurately synchronize internal clocks to a network master clock, or serve as the master clock source.

These adapters contain a dedicated time stamping unit which is driven from a high precision oscillator. Receipt or transmission of a PTP formatted packet triggers the generation of an accurate hardware timestamp which is passed by the adapter to the network device driver. The adapter also allows a PTP stack running on the attached server to discipline the adapter's precision oscillator (both absolute time and clock rate).

4.2 Timestamping Ports

On SFN5322F and SFN6322F adapters, the packet hardware time stamping is limited to PTP packets and only on a single port of the network adapter which is the port closest to the PCIe connector.

The Solarflare Flareon™ SFN7000 series adapters with appropriate AppFlex license support hardware timestamping of all received packets on either adapter interface.

The adapter timestamp function is compliant with the IEEE 1588-2008 (PTP version 2) specifications, and can function as either an 'ordinary' clock or 'master' clock in the network.

Flareon™ SFN7000 Series Dual-Port 10GbE SFP+ Adapters

The SFN7000 series dual-port SFP+ adapters combine ultra low latency with precision time synchronization and hardware timestamping of all received network packets on either physical port of the adapter.

PTP packets can be received on either of the two adapter ports and ports can be configured in an active/standby failover configuration.

The SFN7322F adapter is supplied as a factory-ready PTP adapter - no additional license is required.

Other SFN7000 series adapters can be upgraded with the addition of Solarflare's AppFlex™ Technology license to support PTP and hardware timestamping of all received packets.

A 1PPS bracket kit and cable assembly providing PPS input/output connections can be fitted to the SFN7000 series adapters. Customers interested in the optional PPS kit (Solarflare part number SOLR-PPS-DP10G) should contact their Solarflare sales channel.

For more details of the AppFlex Technology licensing refer to the Solarflare Server Adapter User Guide (SF-103837-CD).

Figure 3: The SFN7000 series Adapter

SFN6322F Dual-Port 10GbE SFP+ Adapter

The SFN6322F is based on the SFN6122F Dual Port SFP+ adapter with additional components for hardware timestamping of PTP packets. The SFN6322F also features a 1PPS input that can be used to calibrate the PTP offset and an extremely accurate 1PPS output timing signal aligned to the adapter's Stratum 3 clock. The SFN6322F combines precision time synchronisation with ultra-low latency 10G Ethernet.

Figure 4: The SFN6322F Adapter

SFN5322F Dual-Port 10GbE SFP+ Adapter

The SFN5322F is based on the SFN5122F Dual Port SFP+ adapter with additional components for hardware timestamping of PTP packets.

Figure 5: The SFN5322F Adapter

Time Synchronization Features

- Ability to maintain synchronization of the system clock typically within 200ns offset from a network master clock. The accuracy obtained is dependent on the PTP master clock, however, the slave adapter clock can be within 50ns offset from the PTP master.
- Stratum 3 compliant oscillator; Oscillator drift < 0.37 PPM per day; < 4.6PPM over 20 years.
- Ability to capture a hardware timestamp as selected frames enter/leave the Ethernet MAC. Time stamping for packets formatted according to IEEE 1588-2008 (PTP version 2).
- Hardware timestamps exposed to Linux via the standard SO_TIMESTAMPING socket API on kernels 2.6.30 later. IOCTL support for time stamping on older kernels, from 2.6.18, that predate SO_TIMESTAMPING.
- Ability to discipline the network adapter's high precision oscillator in response to PTP timing information.
- Support PTP packets over bonded interfaces in an active/standby configuration.
- Support PTP packets over 802.1Q VLAN interfaces.

4.3 Software support

This section describes the software components required to support time synchronization server adapters. To identify the Solarflare net driver and firmware versions used by the Solarflare adapter run the following command where N is the Solarflare interface:

```
# ethtool -i eth<N>
driver: sfc
version: 3.3.0.6246
firmware-version: 3.3.0.6252
bus-info: 0000:04:00.0

# ethtool -i eth<N>
driver: sfc
version: 4.0.2.6628
firmware-version: 4.0.1.6625 rx1 tx1
bus-info: 0000:07:00.0
```

Note: SFN7000 series adapters (the bottom example) have different driver and firmware requirements to 5000/6000 series adapters (the top example).

Firmware

Solarflare Enhanced PTP on SFN5322F or SFN6322F adapters requires a Solarflare adapter with firmware version **3.3.0.6252** or later.

Solarflare Enhanced PTP on SFN7000 series adapters requires a Solarflare adapter with firmware version **4.0.1.6625** or later.

Please refer to chapter 3 Installation on page 20 for instructions on updating the adapter firmware.

Network driver

A Solarflare network adapter driver version **3.3.0.6246** (or later version) is required to support Solarflare Enhanced PTP on SFN5322F or SFN6322F adapters.

A Solarflare network adapter driver version **4.0.2.6628** (or later version) is required to support Solarflare Enhanced PTP on SFN7000 series adapters.

The driver is distributed as a standalone RPM (source and DKMS) or with the OpenOnload distribution.

OpenOnload from version 201210-u1 includes the net driver to support PTP on SFN5322F or SFN6322F adapters.

OpenOnload from version 201310-u1 includes the net driver to support PTP on SFN7000 series adapters.

See Installation on page 20 for more details.

The network driver exposes a number of features of the adapter including:

- Hardware time stamping of PTP formatted packets.

  For kernels prior to 2.6.30 there is no standard interface for supporting hardware packet time stamping and the driver exposes this functionality via an IOCTL interface.

  Starting in Linux kernels 2.6.30, support for hardware time stamping of network packets on TX and RX is formalized and integrated via a new socket option SO_TIMESTAMPING. Details of this interface can be found at http://lxr.linux.no/linux/Documentation/networking/timestamping.txt. Before an application can receive timestamps on a socket it must first issue the IOCTL SIOCSHWTSTAMP to register both the types of packets it wants to receive timestamps on and the format of timestamps it wishes to receive. SIOCSHWTSTAMP is not required by applications using Onload.

  - The SFN5322F|SFN6322F hardware time stamping is limited to PTP formatted packets.
  - SFN7000 series adapters can hardware time stamp all received packets.

- Access to the control of the precision oscillator.

  The adapters contain a precision clock with drift rated to < 1 PPM per year. The driver allows the absolute time and frequency of this clock to be controlled via the Linux PHC subsystem or the proprietary IOCTL interface. The proprietary interface is used by the Solarflare supplied sfptpd stack to run the PTP protocol using the precision oscillator on the adapter.

4.4 Synchronization Model

Figure 6 illustrates the synchronization options supported by sfptpd. The Local Reference Clock, to which all other local clocks are synchronized by sfptpd, is typically a clock on one of the adapters which sfptpd has synchronized to an external PTP master. The Local Reference clock can also be configured as the server system clock disciplined by NTP or from a free-running adapter clock.

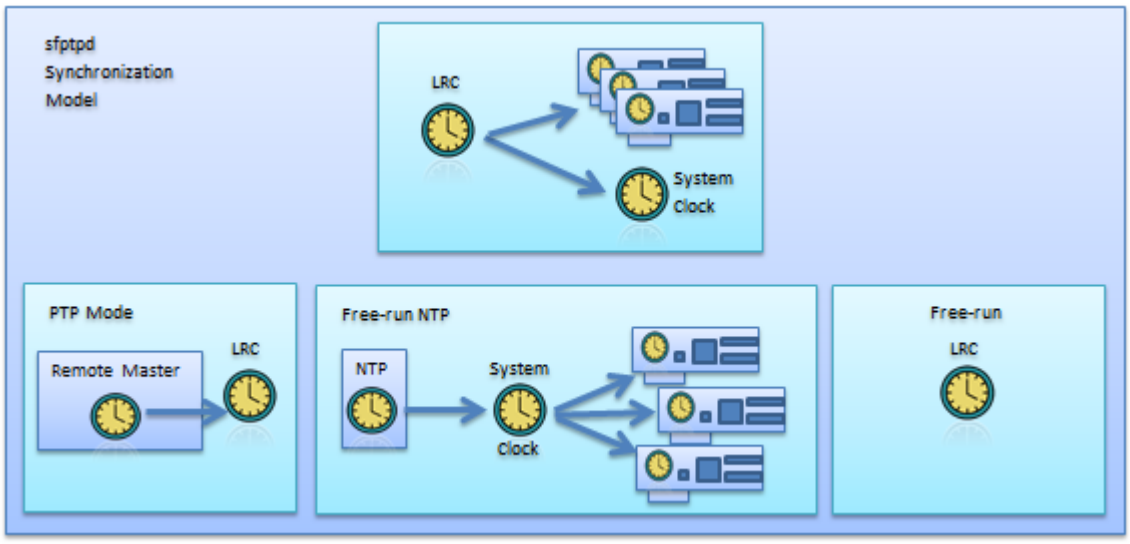

Figure 6: sfptpd Synchronization Model

4.5 Synchronization Modes

The Solarflare Enhanced PTP daemon is able to synchronize multiple local clocks - including the system clock. One local clock is designated the Local Reference Clock (LRC) and can be either synchronized to a remote time source using PTP, or free-running.

Synchronization between the LRC and each local clock is achieved using a clock servo to measure the difference between two clocks and adjusting the local clock to the LRC using a PID filter.

The synchronization mode is selected using the configuration file option:

```
sync-mode ptp                sync-mode freerun
```

PTP Mode

In this mode sfptpd synchronizes the LRC to a remote PTP master clock. With a Solarflare PTP adapter installed, the LRC disciplined by sfptpd is the precision clock on the adapter. sfptpd uses a second clock servo to synchronize the system clock and a clock servo for each additional Solarflare adapter clock in the server.

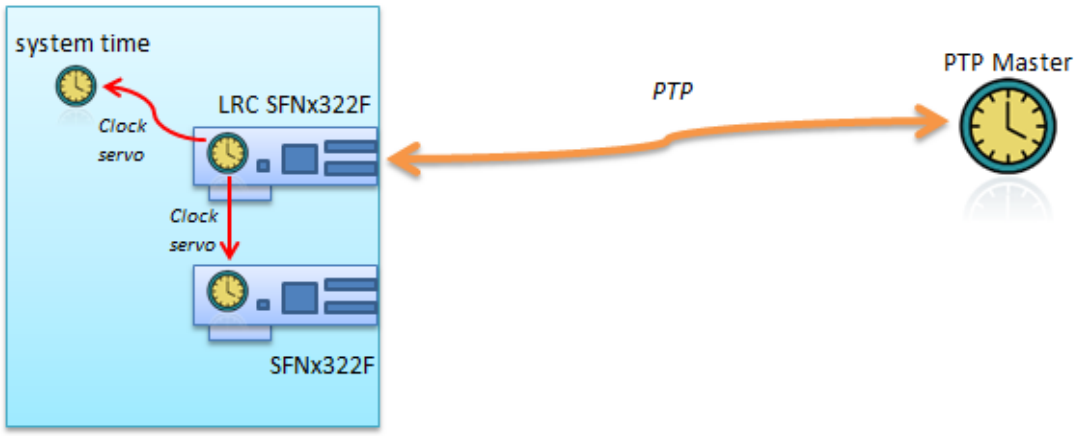

Figure 7: Synchronization Mode - PTP

If there is no Solarflare PTP adapter installed, the LRC is the system clock. sfptpd will discipline the system clock time in this mode.

Free Run Mode

When the `sync-mode` is set to `freerun`, the `freerun-mode` option is then used to identify either nic or ntp:

```
freerun-mode nic          freerun-mode ntp
```

In `freerun-mode nic` the system is NOT being synchronized to a remote clock source i.e. the server is not receiving PTP packets. One local Solarflare adapter clock is selected as the LRC and all other clocks - including the system clock are synchronized to it. Freerun mode nic is illustrated in Figure 8 below.

The system clock is used to set the LRC time once only when sfptpd starts and thereafter the adapter clock runs free.

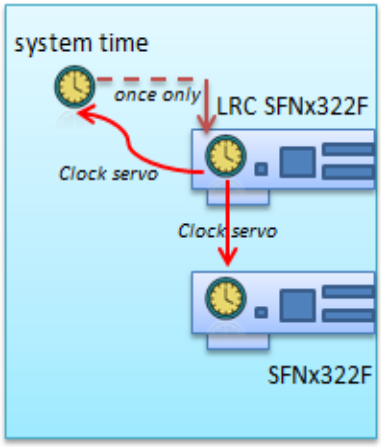

Figure 8: Synchronization Mode - Free Running

sfptpd will discipline the system clock and any additional adapter clocks in this mode.

In freerun-mode NTP, the slave server is running an NTP client allowing the system clock to be synchronized with an external NTP server. All other clocks in the server will then be disciplined from the system clock.

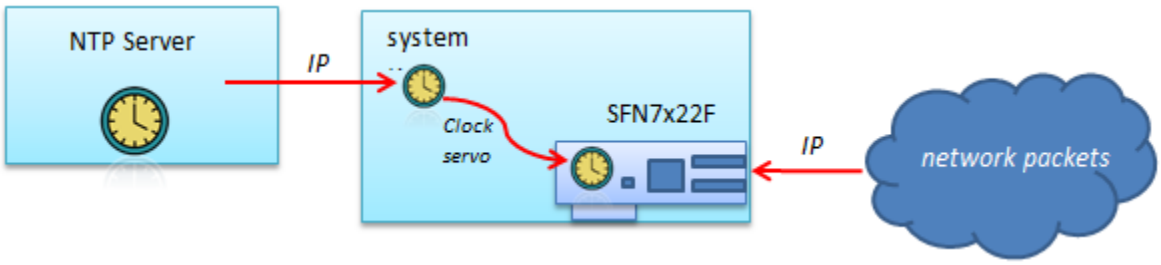

Figure 9: Synchronization Mode - Free Running - NTP

Although sfptpd is running In this mode, the server is not sending or receiving PTP packets. However, sfptpd is synchronizing the clock on the adapter(s) to the system's view of time so hardware timestamps (of non-PTP packets) received from the adapter can be compared with system time. An NTP client can be used in parallel, to synchronize the system clock to an upstream clock reference.

An accurate and stable NTP server should be selected when using this mode.

4.6 Transmission Modes

Multicast Mode

The is the standard PTP mode whereby all PTP packets between master and slave are sent as multicast. The implication is that all slaves receive all Delay_Req, Delay_Resp message pairs between all other slaves and the PTP master clock thereby increasing the amount of traffic on the network. The traffic level generated by multicast transmission is usually not a problem on smaller networks employing only a few PTP slaves.

Multicast mode can be selected for sfptpd using the configuration file option `ptp-network-mode multicast`.

Hybrid Mode

PTP hybrid mode allows a PTP slave clock to use unicast transmission to send Delay_Req messages to the master clock which, in turn, will respond with a unicast Delay_Resp packet direct to the relevant slave. This reduces the level of PTP traffic on the network which can be a factor when scaling to larger networks employing many PTP slaves. Hybrid mode only requires the network to support multicast transmission in one direction - from master to slave.

A sfptpd slave, using hybrid mode, will make three attempts to contact a master clock when sending a Delay_Req message using unicast transmission. If the master clock fails to respond to unicast transmissions the sfptpd slave will revert to multicast transmission. If hybrid mode communication is possible with the master clock, the slave will remain in hybrid mode until/if a new master clock is selected.

sfptpd will generate an error message if the PTP master fails to respond to the unicast Delay_Req message. Error messages will go to stderr or to syslog - depending on the logging configuration.

Hybrid mode is the default mode for sfptpd and can be enabled/disabled using the configuration file option `ptp-network-mode hybrid`.

4.7 Saved Clock Frequency Correction Data

For each Solarflare adapter clock and for the server system clock, sfptpd will save the current frequency correction value every 60 seconds. If the server is rebooted or if sfptpd is restarted, the last saved value is used to continue clock frequency adjustment rather than revert to a zero value which would delay re-synchronization to an external master clock (or re-synchronization to the LRC if the adapter clock is not the active clock receiving from the remote master clock).

Frequency correction files are saved in `/var/lib/sfptpd` - refer to See "PTP State" on page 34. for details.

4.8 Handling of Leap Seconds

Solarflare sfptpd is able to process leap second adjustments. On the occasion of a leap second the sfptpd daemon will suspend clock synchronization, step the clock currently being disciplined by one second and then recommence normal synchronization when the next announce message is received.

4.9 System Time

Applications running on the server can call standard POSIX/Linux system time calls such as `clock_gettime()` to obtain an accurate time reading. Most modern Linux kernels implement these calls entirely at user space (using a VDSO) and provide an accurate time access API with low CPU overhead.

On a Xeon® CPU E3130 @ 3.20GHz machine, Solarflare have measured the execution time of calls to `clock_gettime()` as 61ns when making a syscall (kernel.vsyscall64=0), and 24ns when configured to use a VDSO (kernel.vsyscall64=1).

NOTE: Although it would seem useful for applications to have the ability to read the time directly from the clock on the Solarflare adapter, this would require hardware reads across the PCIe bus. PCIe reads are slow and such a call would take in the order of a microsecond to complete and would therefore introduce significant errors in the time obtained.

4.10 Loss of PTP Link Network Connection

If the network connection to the external master clock is lost at any point, Solarflare's sfptpd continues to discipline all clocks in the system including the system clock.

sfptpd maintains a clock frequency correction file for each clock in a server. If the LRC is a Solarflare PTP adapter, sfptpd will continue to discipline the LRC using the LRC frequency correction file. sfptpd will ensure that other clocks, hardware or system, will continue to synchronize to the LRC.

When there is no Solarflare PTP adapter in the server - or if the LRC is the system clock, sfptpd will continue to discipline the system clock using the system clock frequency correction file.

Following restoration of the network link, PTP packets will be received and sfptpd will resume normal discipline procedure.

To ensure that saved frequency correction files are used to discipline clocks, the configuration file option `persistent-clock-correction` should be enabled.

Chapter 5: Installation

5.1 System Requirements

This section identifies all components required to deploy the Solarflare PTP adapter for operation.

Supported Linux Kernel Versions

Solarflare Enhanced PTP is supported on the following OS/kernels versions:

- Linux® 2.6 and 3.x Kernels (32 bit and 64 bit) for the following distributions: RHEL 5, 6 and MRG. SLES 10, 11 and SLERT.

Note: Linux kernels prior to 2.6.30 can only deliver microsecond resolution.

Other Requirements

- SFN7322F, SFN7122F, SFN5322F or SFN6322F network server adapter.
- SFN5322F/SFN6322F adapters must have a minimum firmware version 3.3.0.6252 for sfptpd.
- SFN5322F/SFN6322F adapters must have a minimum driver version 3.3.0.6246 for sfptpd.
- SFN7000 series adapters must have a minimum firmware version of 4.0.1.6625 for sfptpd.
- SFN7000 series adapters must have a minimum driver version of 4.0.2.6628 for sfptpd.
- Solarflare supplied sfptpd stack to use Solarflare Enhanced PTP.

Verify Solarflare Adapter Driver and Firmware Versions

To check the Solarflare adapter driver and firmware versions use the Linux ethtool command e.g.

```
# ethtool -i eth<N>
```

If the driver and firmware versions do not meet the minimum versions required for Solarflare sfptpd refer to the sections below for upgrade procedures.

Step 1: Server Pre-Install Setup

Tickless Kernel - nohz

A feature of modern operating systems is that they use a "tickless" kernel which aims to reduce power consumption during kernel idle periods. This is achieved by stopping the regular timer tick on CPU cores which are idle. However, experiments at Solarflare have proven that PTP produces improved and more consistent results when the kernel always receives periodic timer ticks.

PTP relies on the ability to accurately change the speed of the system clock by very small and precise amounts. The Linux kernel implements this adjustment to system clock rate with integer arithmetic, minimizing the error term to the target clock rate in every timer tick. However, when the timer tick doesn't run, the error in tracking to the requested clock rate increases, and the system time diverges from the clock rate requested. When the system wakes from idle, the timer tick runs and the kernel corrects for the error term.

Whether the kernel operates in a "tickless" mode is configured by the "nohz" boot time option with the majority of Linux distributions defaulting to a tickless kernel. To achieve the highest accuracy with PTP, Solarflare suggest configuring the kernel to receive timer ticks even when the system is idle. This can be achieved by adding "`nohz=off`" to the kernel boot parameters in the `/boot/grub/grub.conf` file.

IPTables

Users must ensure that no rule exists in iptables that will prevent PTP packets from reaching the slave sfptpd process.

Step 2: Install the Network Adapter

Complete instructions for the deployment and installation of the network adapter can be found in the Solarflare Server Adapter User Guide SF-103837-CD.

Step 3: Install Network Driver

Download and install the driver DKMS RPM

The DKMS system must be installed before the Solarflare DKMS RPM package. The following command can be used to verify DKMS support and version number.

```
# dkms --version
```

Refer to the driver release notes for instructions to install or up date the dkms version if required.

1. Download the driver zip file SF-104979-LS from https://support.solarflare.com/.
2. Copy the zipfile to a directory on the target machine e.g. `/tmp`, unzip it and, as root, execute the following commands:

```
# rpm -ivh sfc-dkms-3.3.0.6246-0.sf.1.noarch.rpm
```

3. Load the network adapter driver.

```
# modprobe sfc
```

Step 4: Update Adapter Firmware

NOTE: The Solarflare Utilities RPM for Linux contains a boot ROM utility (sfboot), a flash firmware update utility (sfupdate) and a AppFlex license upgrade utility (sfkey).

The RPM package is available as a 32bit binary and 64bit binary:

SF-105095-LS is a 32bit binary

SF-107601-LS is a 64bit binary

1. Download the sfutilities package from https://support.solarflare.com/.
2. Unzip the file to reveal the binary RPM
3. Install the RPM e.g.

```
# rpm -Uvh sfutils-<version>.rpm
```

4. Identify the current firmware version on the adapter.

```
# sfupdate
```

5. Replace the adapter firmware with the version in this sfupdate.

```
# sfupdate --write
```

Full instructions on using sfupdate, sfboot and sfkey can be found in the Solarflare Network Adapter User Guide SF-103837-CD.

Step 5: Identify the Timestamping Port

Connect the SFN7000 series Timestamping Port

The Solarflare 7000 series adapters support hardware timestamping of all packets on either port of the adapter.

Ports on the same adapter, or from different adapters in the same server, can be bonded in an active/standby failover configuration.

Connect the SFN5322F|SFN6322F Timestamping Port

Hardware time stamping is only functional on a single port of the adapter which is the port closest to the PCIe connector.

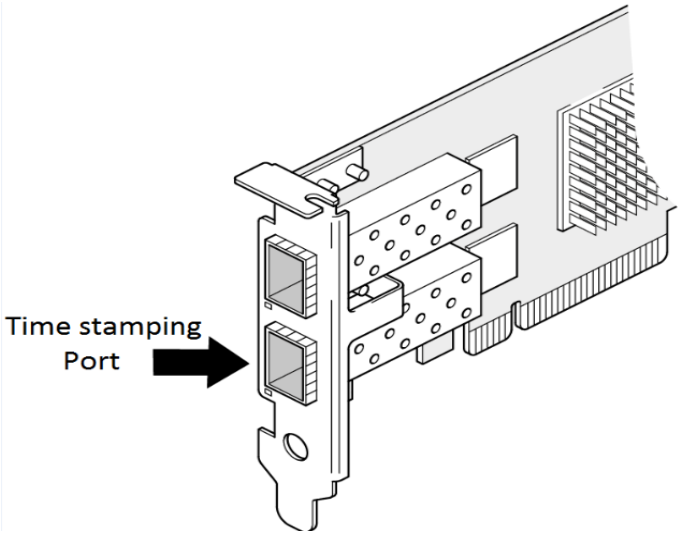

Figure 10: Identify Timestamping Port

Use ethtool to identify the hardware timestamping port:

```
ethtool -i eth<N>
driver: sfc
version: 3.3.0.6246
firmware-version: 3.3.0.6252
bus-info: 0000:07:00.0
```

From the PCI bus-info a zero function value (last digit) identifies the timestamping port on the network adapter.

It can be useful to use ethtool to identify the timestamping port on the back panel by 'blinking' the port LED for a specified number of seconds:

```
ethtool -p eth<N> 10
```

Chapter 6: Using sfptpd

6.1 Download sfptpd

The Solarflare Enhanced PTP, sfptpd, is a PTP daemon adapted by Solarflare to work with Solarflare time synchronization server adapters. The sfptpd daemon is an implementation of IEEE-1588-2008 (PTP version 2) and is not compatible with IEEE- 1588-2002 (PTP version 1).

NOTE: The sfptpd package is available as a 32bit binary and 64bit binary:

SF-108909-LS is a 32bit binary

SF-108910-LS is a 64bit binary

1. Download the required sfptpd package from https://support.solarflare.com/.
2. Unpack the compressed file using the tar command e.g.

```
tar -zxvf SF-108910-LS-<version>.tgz
```

6.2 Distribution Package

The Enhanced PTP distribution contains the following files:

- Copyright notice
- Release notes
- sfptpd daemon
- Config directory
  - ptp_master.cfg, an example configuration file for PTP master mode using PTP synchronization mode.
  - ptp_slave.cfg, an example PTP slave configuration file.

6.3 Run sfptpd

NOTE: sfptpd can take 15-30 minutes to initially stabilize the times on the slave machines.

- To view all sfptpd configuration file options run the following command.

```
./sfptpd -h
```

- Ensure that the timestamping port of the adapter is configured with an IP address suitable for the network configuration.

Run sfptpd as PTP Master

- To start the sfptpd process as the PTP master using the default `ptp_master.cfg` file in the config sub-directory

```
./sfptpd -ieth<N> -fconfig/ptp_master.cfg
```

Run sfptpd as PTP Slave

- To start the sfptpd process as a PTP slave using the default `ptp_slave.cfg` file in the config sub-directory

```
./sfptpd -ieth<N> -fconfig/ptp_slave.cfg
```

Run sfptpd as PTP Slave - freerun mode

- To start the sfptpd process as a PTP slave using the default `freerun.cfg` file in the config sub-directory

```
./sfptpd -ieth<N> -fconfig/freerun.cfg
```

Where N is the identifier of the timestamping port on the adapter.

6.4 PTP over VLAN

Solarflare Enhanced PTP supports PTP packets over tagged 802.1Q Virtual Local Area Network (VLAN) interfaces. Users should consult the relevant OS documentation for VLAN configuration instructions. Assuming interface eth2.120 is a network interface configured with VLAN tag 120, the following example identifies sfptpd VLAN configuration.

```
./sfptpd -ieth2.120 -fconfig/ptp_slave.cfg
```

6.5 PTP over Bonded Interfaces

Solarflare adapters and sfptpd support PTP packets over bonded interfaces in an active/standby mode. Bonding of Solarflare interfaces employs the Linux bonding driver. Multiple ports can be included into a single bond where one port is selected as the active interface and all others are standby.

- sfptpd will detect which port is active and which ports are passive in the bond.
- sfptpd will discipline the high precision clock on the active port's network adapter.
- sfptpd will discipline the clocks of passive ports from the active adapter's clock.
- Via the bonding driver the user can select the active port (and therefore clock).
- A bond can include non-PTP capable Solarflare ports - sfptpd will switch to software time-stamping when a non-hardware time-stamping port becomes active.

- A bond can include non Solarflare ports - sfptpd will switch to software time-stamping when a non-Solarflare port becomes active.
- A bond can include any number of ports.

Bonding Configuration

Bonding of Solarflare interfaces is handled by the standard Linux bonding driver. Users should refer to http://www.kernel.org/doc/Documentation/networking/bonding.txt for details of alternative methods for bonding configuration. The following example is a manually bonding configuration using ifenslave:

```
# modprobe bonding miimon=100 mode=1 xmit_hash_policy=layer2 primary=eth5
# ifconfig bond0 172.16.136.27/21
# ifenslave bond0 eth0 eth1
```

To run sfptpd over the bonded interfaces:

```
./sfptpd -ibond<N> -fconfig/ptp_slave.cfg
```

Action on Active port Failover

The active port in the bonding interface identified on the command line with the -i option is the active clock. In the event of failure of the active port:

If the standby port is a Solarflare PTP capable port, synchronization will continue as before because the standby port's clock is kept in sync by sfptpd with the active port's clock. If the standby port is a non-Solarflare adapter or a Solarflare port that does not support hardware timestamps, then the system clock becomes the LRC and sfptpd uses software timestamping.

Each clock has its own freq-correction file, updated every 60 seconds, which is used to record the frequency correction PPB value needed to keep the clock in sync with the LRC in the event of an sfptpd restart or a server reboot.

6.6 Hardware Timestamps

This feature is available for the SFN7000 series adapters only.

On Solarflare SFN7000 series adapters sfptpd can be used to enable hardware timestamping of all packets (to the Linux kernel) on specified interfaces. Interfaces are identified as a list using the following configuration file option:

```
timestamping-interfaces  [<name | mac-address | *>]
```

- To timestamp all received packets on all interfaces:

```
timestamping-interfaces *
```

- To timestamp all received packets on eth2 and eth3:

```
timestamping-interfaces eth2 eth3
```

6.7 Hardware Timestamps Enable/Disable

This feature is available for the SFN7000 series adapters only.

- To enable/disable hardware timestamping of all received network packets after sfptpd exits, use the following configuration file option:

```
timestamping-disable-on-exit [<off | on>]
```

6.8 Hardware Timestamps (Kernel/Onload)

Using the SFN7000 series adapters, applications can recover hardware timestamps for all received packets using the SO_TIMESTAMPING socket option. For more details of hardware packet timestamps when using the kernel driver see the Solarflare Server Adapter User Guide (SF-103837-CD). Fore more details of using hardware packet timestamps when using OpenOnload see the Onload User Guide (SF-104474-CD).

6.9 sfptpd in Operation

The following charts show sfptpd performance when the Solarflare adapter is configured in PTP slave mode to a 3rd party Grandmaster clock via a network switch. In this example Ethernet interface 4 (eth4) is the interface receiving PTP packets and sfptpd is disciplining the system clock.

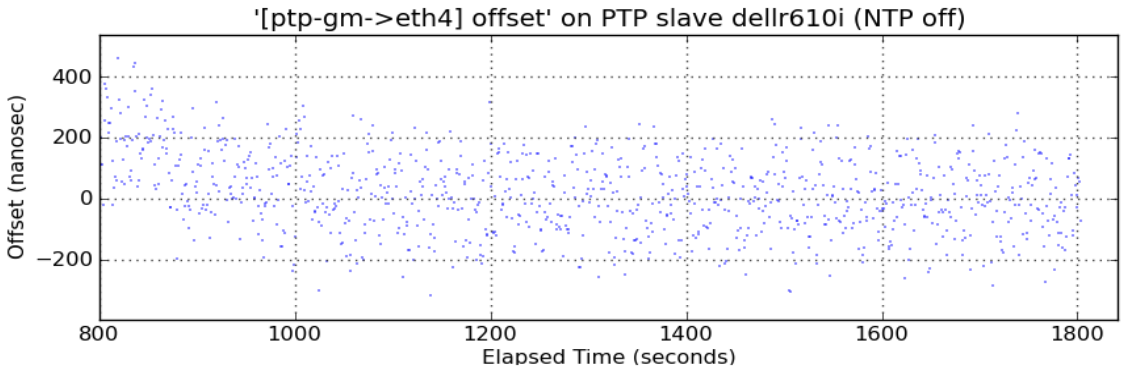

Figure 11: Offset of adapter's clock to the PTP master

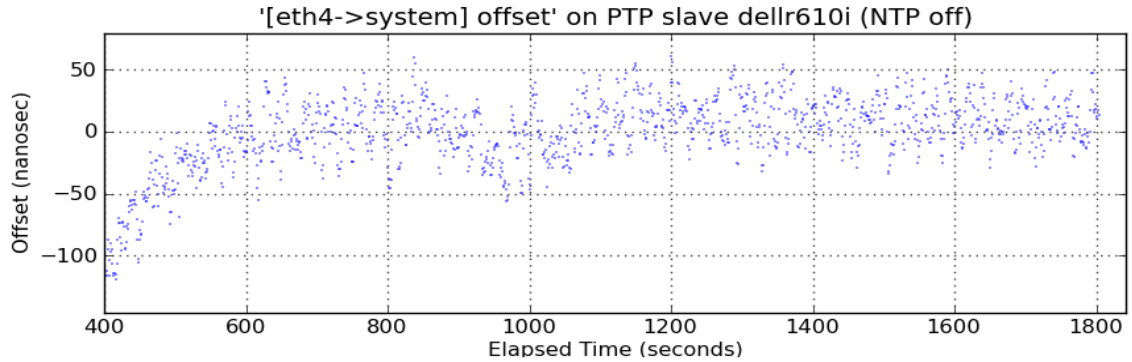

Figure 12: Offset of the system clock to the adapter's clock

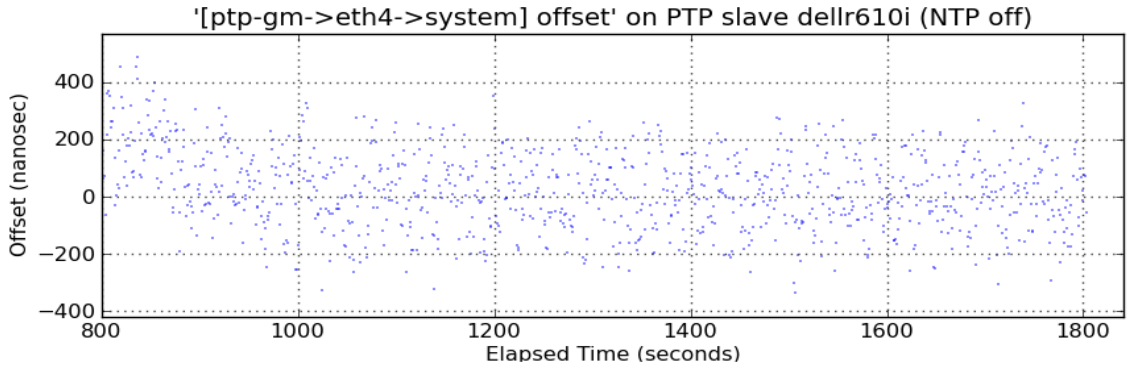

Figure 13: Offset of the server's system clock to the PTP master

The following charts show sfptpd performance when two Solarflare PTP adapters are present and configured in a bonded interface in a PTP slave server. Arrows on the chart identify what happens during bond failover and failback events.The upper chart shows the consistent offset of the server's system clock from the clock on the active port during repeated bond failover and failback events. For the same period, the middle chart shows the offset of the active adapter clock from the external grandmaster during the failover and failback of the bonded ports. For the same period the lower chart shows the offset of the server system clock from the external PTP master clock.

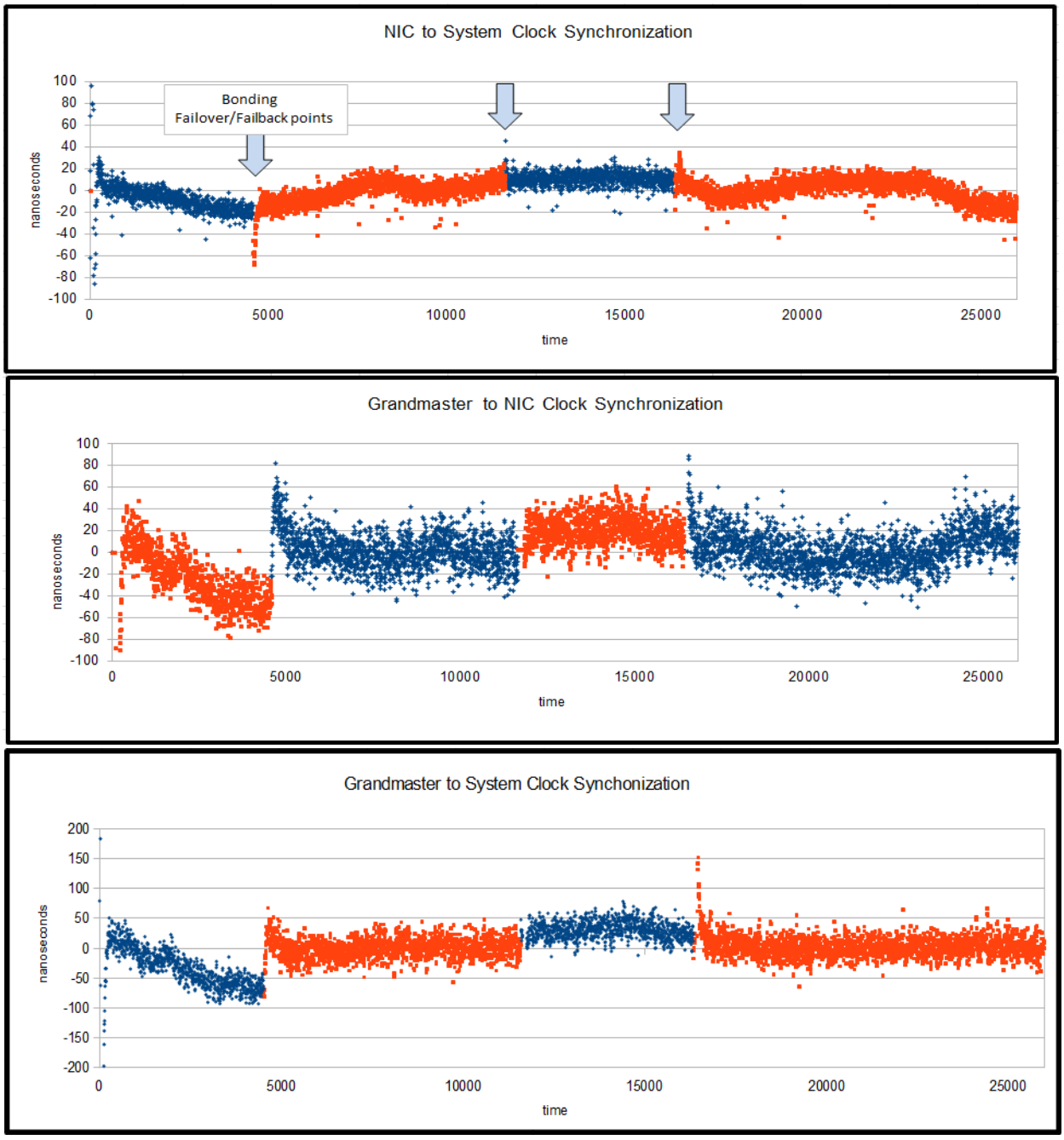

Figure 14: Bonding failover Performance

6.10 Accuracy under Network Load

To obtain the highest accuracy the PTP protocol requires a network with constant latency. Standards such as “PTP boundary clock” and “PTP transparent clock” allow network switches to be PTP aware and measure latencies to allow the PTP end points to compensate for any variance in switching times for PTP packets. However, even with standard non-PTP aware switches, the two stage PTP synchronization approach used by the adapter can provide good accuracy under significant network load.

Solarflare has demonstrated slave to master offsets within 200ns on a lightly loaded network. However, even under bursty conditions of up to 50% 10G line rate, the SFN5322F|SFN6322F demonstrated slave to master offsets of within 500ns. When the bursty condition cleared, the slave to master offsets returned to within 200ns.

Figure 15 shows SFN5322F PTP accuracy when used in an environment with bursty network load of up to 50% line rate. The test employed the SFN5322F adapters as master and slave configured via a Cisco Nexus 5000 series switch.

Network load tests, producing similar results, have been repeated with the SFN6322F and SFN7000 series adapters.

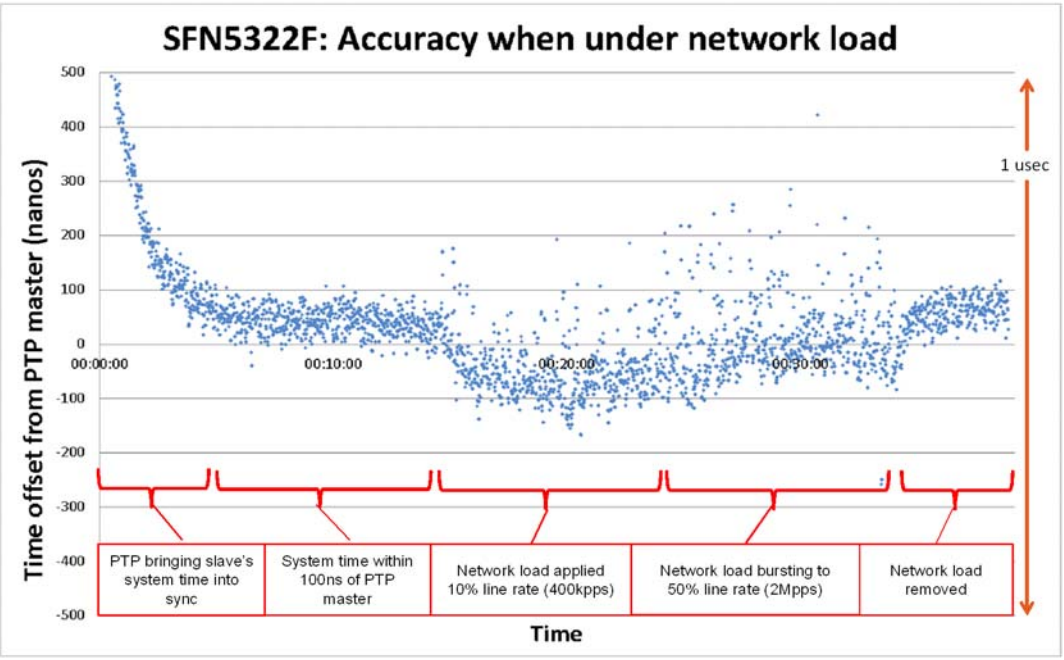

Figure 15: PTPd Under Load

Chapter 7: Configuration Files

7.1 Overview

The sfptpd distribution unpacks default configuration files into the config sub-directory. Default options for master and slave modes are enabled within each config file.

Within the config files lines beginning with a # symbol (commented out) are ignored. Additional or different options can be selected by un-commenting the option line and, if required, entering a different value for the option e.g.

- Commented out option lines will be ignored e.g.

```
#ptp-announce-interval 1
```

- To enable an option un-comment the line and change the value if required e.g.

```
ptp-announce-interval 3
```

The user is free to create additional configuration files and store these anywhere on the local server. Configuration files can have any name, but must have an .cfg file extension and the full path to the file must be identified on the sfptpd command line.

Quickstart Default Master Configuration

To run sfptpd on a server in default master mode use the ptp_master.cfg file located in the config sub-directory.

```
./sfptpd -ieth<N> -fconfig/ptp_master.cfg
```

Table 4 identifies the default PTP options configured when using the ptp_master.cfg file.

Table 4: Default PTP master options

Option	Default Value	Description
sync-mode	ptp	sfptpd synchronizes the LRC to a remote PTP master clock. With a Solarflare PTP adapter installed, the LRC disciplined by sfptpd is the precision clock on the adapter. sfptpd uses a second clock servo to synchronize the system clock and a clock servo for each additional Solarflare adapter clock in the server. When no Solarflare PTP adapter is present, the LRC is the system clock.
ptp-network-mode	hybrid	Delay_Req and Delay_Resp messages are sent as UDP unicast - all other PTP messages are multicast.
persistent-clock-correction	on	Periodically saved clock frequency corrections are used to discipline local clocks.

Table 4: Default PTP master options

Option	Default Value	Description
ptp-mode	master	PTP master clock mode.
ptp-stats	off	Controls the level of stats logging to stderr or file.
ptp-tx-latency	0	Outbound latency in nanoseconds.
ptp-rx-latency	0	Inbound latency in nanoseconds.
ptp-ttl	1	TTL value in transmitted PTP packets.
ptp-domain	0	PTP domain.

Quickstart Default Slave Configuration

To run sfptpd on a server in default slave mode use the ptp_slave.cfg file which is located in the config sub-directory.

```
./sfptpd -ieth<N> -fconfig/ptp_slave.cfg
```

Table 5 identifies the default PTP options configured when using the ptp_slave.cfg file.

Table 5: Default PTP slave options

Option	Default Value	Description
sync-mode	ptp	sfptpd synchronizes the LRC to a remote PTP master clock. With a Solarflare PTP adapter installed, the LRC disciplined by sfptpd is the precision clock on the adapter. sfptpd uses a second clock servo to synchronize the system clock and a clock servo for each additional Solarflare adapter clock in the server. When no Solarflare PTP adapter is present, the LRC is the system clock.
message-log	stderr	Direct PTP event messages to stderr.
ptp-network-mode	hybrid	Delay_Req and Delay_Resp messages are sent as UDP unicast - all other PTP messages are multicast.
persistent-clock-correction	on	Periodically saved clock frequency corrections are used to discipline local clocks.
ptp-mode	slave	PTP slave clock mode.
ptp-stats	off	Controls the level of stats logging to stderr or file.
ptp-tx-latency	0	Outbound latency in nanoseconds.
ptp-rx-latency	0	Inbound latency in nanoseconds.

Table 5: Default PTP slave options

Option	Default Value	Description
ptp-ttl	1	TTL value in transmitted PTP packets.
ptp-domain	0	PTP domain.

7.2 Command Line options

sfptpd supports a limited number of command line options which override the equivalent config file options. Use the `sfptpd -h` command to identify supported command line options.

7.3 Starting sfptpd with a config file

To have sfptpd configured from a config file, the full path to the config file location must be identified on the sfptpd command line, for example, to use the default slave config file in the config sub-directory:

```
./sfptpd -ieth<N> -fconfig/ptp_slave.cfg
```

```
./sfptpd -ieth<N> -fconfig/ptp_master.cfg
```

7.4 Additional Options

The default configuration files do not contain all sfptpd configuration file options. To see the complete list of config file options run the **`sfptpd -h`** command.

Add the required options to the default configuration files or to user-created configuration files.

Chapter 8: PTP State

8.1 View Statistics Files

On a server using the Solarflare sfptpd package, PTP alarms, status and performance data is accumulated in files created in the following directory:

```
/var/lib/sfptpd
```

From these files the user is able to monitor the performance and status of sfptpd and the PTP server.

8.2 The Toplogy File

When viewed on a sfptpd slave server, the topology file presents a PTP clock hierarchy diagram showing all clocks local to the slave server and PTP network elements between the slave and master clock. A PTP Boundary Clock would be visible in the file as a parent to the slave. A PTP Transparent Clock would not be visible in the topology file.

The topology file identifies the current state of the slave clock, the interface being used to receive PTP messages and the timestamping mode being used. Each clock in the topology is identified by its UUID which is derived from its MAC address. Nanosecond values between clocks are the offset values recorded during the last file update.

The following output is a example of a topology file showing alarm states.

```
state: ptp-slave-alarm
alarms: no-sync-pkts no-delay-resps
interface: eth2 (eth2)
timestamping: hw
====================

          grandmaster
      000f:53ff:fe16:0474/1
                |
                |
           -170.000 ns
                |
                |
                v
        phc0(eth2/eth3)
       000f:53ff:fe21:9bb0
                |
                |
            -52.688 ns
                |
                |
                v
             system
             system
```

In the above example, the slave is in an alarm state indicating that Sync messages and Delay_Response messages are not currently being received from the master clock. At the same time the output from sfptpd would highlight the one-way-delay and offset values in red text to indicate a problem in the PTP network.

The topology file can be periodically monitored, for example, using a script to extract key fields, to monitor the current connection state and synchronization status of the PTP slave.

NOTE: During normal operation the topology file is updated every second. However, alarm or state changes are reflected in the file immediately.

8.3 Statistics Files.

Of all the files created, the `freq-correction-*` files are persistent and will be preserved over sfptpd restart and over server reboot. All other files are non-persistent and are created when sfptpd is started.

Table 6 lists statistics files created by sfptpd.

Table 6: Statistics Files Created by sfptpd

File name	Persistent	Description
freq-correction-<UUID>	YES	Contains the frequency correction value used to discipline the clock.
freq-correction-system	YES	Contains the frequency correction value used to discipline the server system clock.
state-system	NO	The contents of the this file depend of the current PTP mode. In slave mode this file contains historical offset and status data for the slave server.
stats-system	NO	Contains accumulated PTP performance data for the server including counts of the PTP message types sent and received.
state-<UUID>	NO	Contains status information relevant to the clock identified by the UUID identifier.
stats-<UUID>	NO	Contains accumulated synchronization data for the clock identified by the UUID identifier.
topology	NO	See above for an in depth description of this file.

The following tables identify and describe sfptpd stats files:

Table 7: FILE: freq-correction-<UUID>

Name	Description
freq-correction-<UUID>	This identifier is constructed from the hardware address of the clock port.
value	This is the frequency correction value used to discipline the clock. The value is updated once per minute when the clock is in sync with its synchronization time source. This file persists over server reboot and sfptpd restart. Following either event, the frequency correction value is used to recommence disciplining of the clock ensuring faster re-convergence.

Table 8: FILE: state-<UUID>

Name	Description
NOTE: the contents of this file depend on the current PTP mode and position of the clock in the PTP hierarchy. A state file exists for each adapter clock and for the server system clock.	
clock-name	identify the clock using this clock-id.
clock-id	the clock UUID.
state	ptp_slave | ptp_master also identifies if the clock is subject to any current alarms.
interface	identifies the PTP clock interface.
timestamping	current timestamping mode.
offset-from -master	Offset (nanoseconds) of LRC from the master clock.
one-way-delay	One-way-delay (nanoseconds) between LRC and master clock.
freq-adjustment-ppb	Current frequency correction value used to discipline this clock.
observed-drift	Drift in nanoseconds of slave LRC to master clock.

Table 8: FILE: state-<UUID>

Name	Description
in-sync	0 observed offset > 1 microsecond 1 observed offset < 1 microsecond
steps-removed	steps removed from the master clock.
parent-clock-id	UUID of the PTP parent clock. If there is a boundary clock between LRC and the master clock this will identify the boundary clock.
parent-port-num	Port number relayed to the server by the master clock in the SourcePortID parameter.
grandmaster-id	The UUID grandmaster clock constructed from the grandmaster hardware address.
grandmaster-clock-class	The current Grandmaster class value.
grandmaster-clock-accuracy	The current Grandmaster accuracy value.
grandmaster-priority1	The current Grandmaster priority 1 value.
grandmaster-priority2	The current Grandmaster priority 2 value.
current-utc-offset	The current UTC offset in seconds from TAI value specified in the config file with the **ptp-utc-offset** value.
leap-59	1 indicates a leap second is scheduled.
leap-61	1 indicates a leap second is scheduled.

Table 9: FILE: stats-<UUID>

Name	Description
NOTE: the contents of this file depend on the current PTP mode and position of the clock in the PTP hierarchy. A state file exists for each adapter clock and for the server system clock.	
offset-from-master freq-adjustment-ppb one-way-delay	mean, min and max values are accumulated for these statistics. The number of samples taken during each period is recorded as is the start/end time of each sample period.

Table 9: FILE: stats-<UUID>

Name	Description
The stats file retains accumulated counts of each PTP message type sent or received from the server. The statistics file also records the number of PTP packets sent of received without being hardware timestamped. The number of samples taken during each period is recorded as is the start/end time of each sample period.	

Chapter 9: Pulse Per Second (1PPS)

9.1 Asymmetric Networks

Asymmetric networks present a particular problem when attempting to account for network latency during PTP offset calculations between master and slave servers. PTP assumes symmetry in the network and the PTP protocol is not able to detect asymmetry in the network paths between master and slave.

Asymmetry can be present for a number of reasons including the store and forward delay in switches serving asymmetric networks as illustrated in Figure 16.

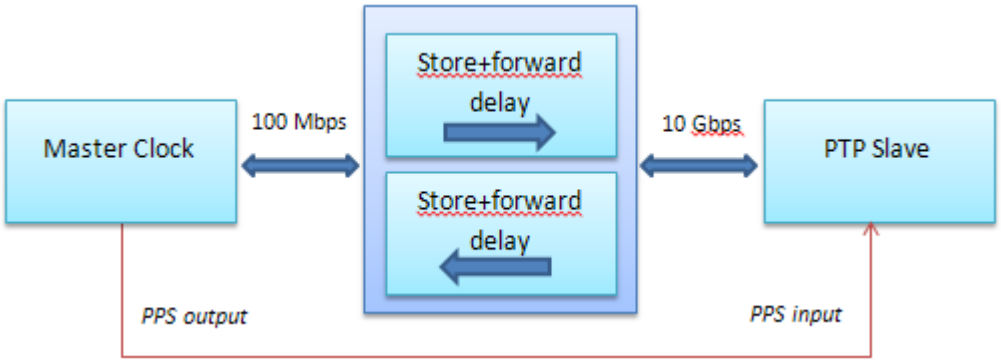

Figure 16: Asymmetric Network

The result is that PTP offsets between master and slave will converge, but will be wrong by a constant offset from the master equal to half the asymmetry, for example

```
Transmit time master to slave:   5us
Transmit time slave to master:   1us
                One way delay:   (5+1)/2 = 3us
So 3us is added to the time offsets received from the master clock.
1PPS will display a mean offset value of 2us (5-3)
Actual asymmetry should be double this observed value i.e. 4us.
```

Measuring and Adjusting for Asymmetric Latency

The Solarflare SFN6322F and SFN7000 series adapters supports 1PPS input/output interfaces[1] to allow asymmetry in the network to be measured. On a dedicated wire connection between master 1PPS output and slave 1PPS input, the master emits a single pulse every second. The leading edge of each pulse denotes the exact start of a one second period. When the leading edge of a pulse is detected by the slave adapter, firmware on the adapter is able to calculate the offset from its own 'start of second period'.

1. The SFN6322F adapter is factory fitted with 1PPS I/O connectors. The Solarflare SFN7000 series adapters require the optional PPS bracket kit and cable assembly (product code SOLR-PPS-DP10G) available from Solarflare sales channels.

If the initial observed mean 1PPS offset value is a negative value, it means the master->slave path is slower than the slave->master path, therefore the **ptp_rx_latency** configuration file option on the slave server is used to compensate the receive latency.

If the initial observed mean 1PPS offset value is a positive value, it means the slave->master path is slower than the master->slave path, therefore the **ptp_tx_latency** configuration file option on the slave server is used to compensate the transmit latency.

This 1PPS calibration is only required once when configuring the network and need only be performed on one slave server in each network segment which share a common network path to the PTP master. There is no need for a permanent 1PPS connection to the Solarflare adapter. Refer to 1PPS in Practice on page 42.

9.2 1PPS Measurement Procedure

1. sfptpd should be running between master and slave servers, and should be synchronized before the 1PPS value is measured and applied.
2. The master 1PPS output should be connected to a single slave 1PPS input.
3. On the slave server, for a short period e.g. 5 minutes, observe the 1PPS mean offset value from the `ptp_mc_pps_off_mean` file to identify the mean offset value. Refer to Appendix B: 1PPS Statistical Data on page 47 for instructions on reading the 1PPS statistical data files.
4. On all slaves on the same network segment, configure sfptpd with knowledge of the mean 1PPS offset.

- If the initial observed 1PPS offset is a negative value, then all subsequent offsets should be added as positive values to the ptp_rx_latency option. The ptp_tx_latency option in this case should be zero.
- If the initial observed 1PPS offset is a positive value, then all subsequent offsets should be added as positive values to the ptp_tx_latency option. The ptp_rx_latency option in this case should be zero.

5. Continue to observe the 1PPS compensated mean offsets.
6. Repeat steps 3-5 adding or subtracting the 1PPS mean offset (doubled) value each time to the last applied value until the observed 1PPS mean value is as close to zero as possible.

1PPS asymmetric compensation examples:

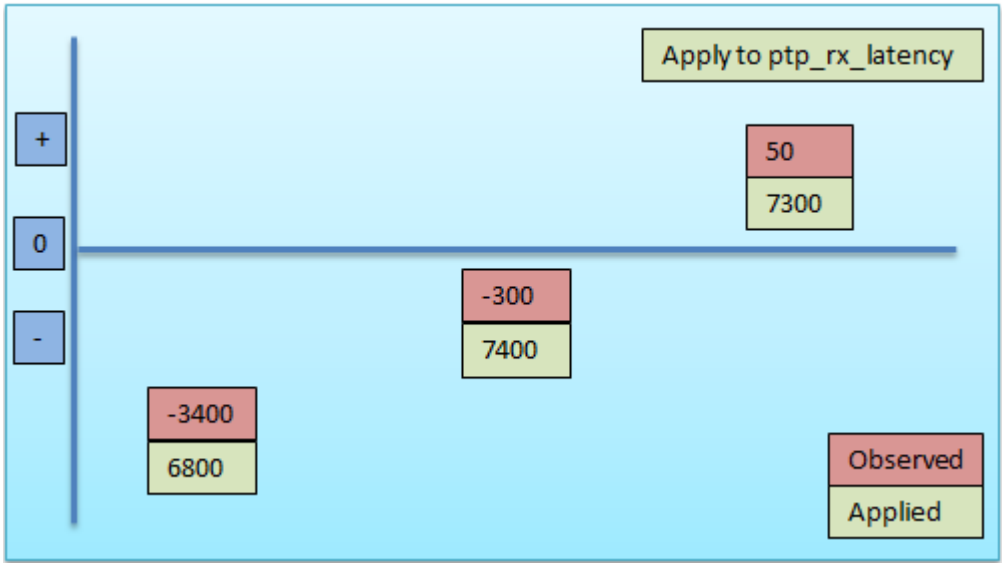

Figure 17: 1PPS - Example

In the above example the initial observed 1PPS offset is -3400. This value is doubled and applied to sfptpd using the ptp_rx_latency parameter as 6800.

sfptpd is restarted and the next observed 1PPS offset is -300, this value again is doubled and applied to the original compensation value (6800 + 600 = 7400).

sfptpd is restarted and the final observed 1PPS offset is +50 meaning the previous compensation value caused sfptpd to over-compensate, so the +50 is doubled and subtracted from the previous compensation value (7400 - 100 = 7300).

9.3 1PPS in Practice

The following sections demonstrate the effect of applying the 1PPS mean offset value to sfptpd.

1PPS Measurements - Asymmetric Network

Figure 18 shows the 1PPS offsets observed on an asymmetric network consisting of a 3rd party grandmaster clock (100Mbps interface) and Solarflare 10Gbps (slave) adapter connected via a standard network switch.

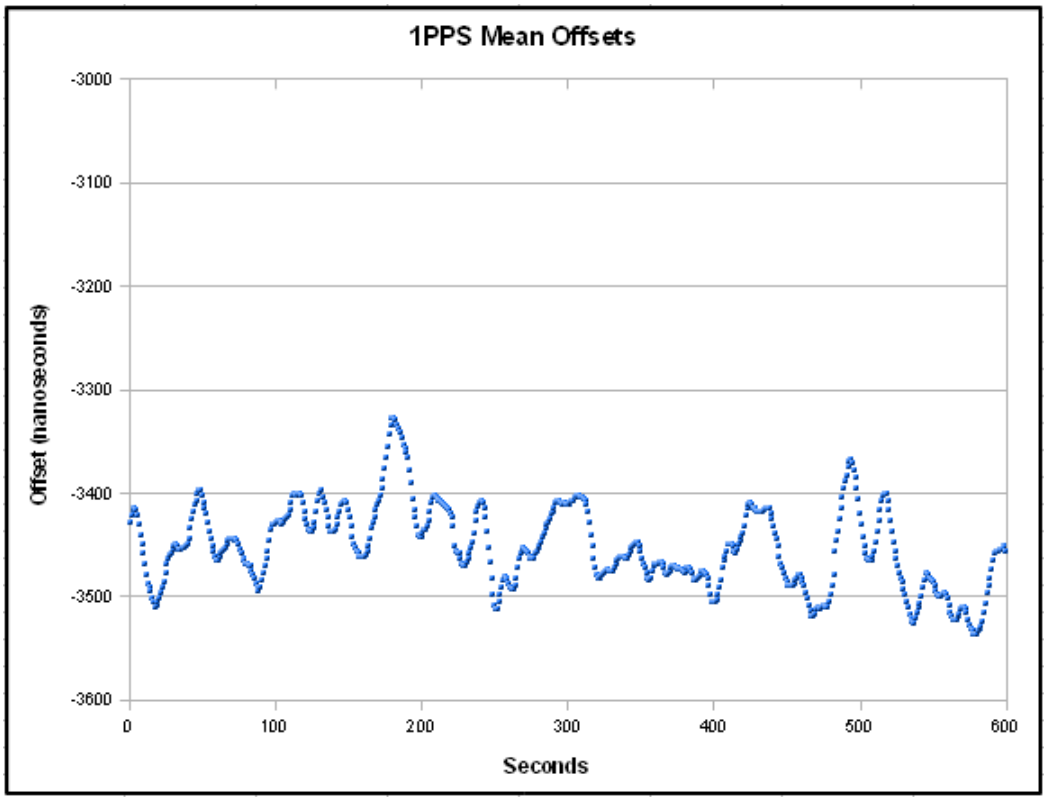

Figure 18: 1PPS Measurement - Asymmetric

1PPS Measurements - Identify Mean Offset

In this particular instance the 1PPS offset is observed for a period of 10 minutes before the mean offset value is identified as `ptp_mc_pps_off_mean: -3450.` Refer to Appendix B: 1PPS Statistical Data on page 47 for instructions on reading the 1PPS statistical data.

1PPS Measurements - Apply Mean Offset

The 1PPS mean offset value **should be doubled** and applied to sfptpd via the slave server configuration file as follows e.g.

```
ptp_rx_latency 6900 ptp_tx_latency 0
```

Figure 19 demonstrates 1PPS output after the (doubled) mean offset has been applied to sfptpd.

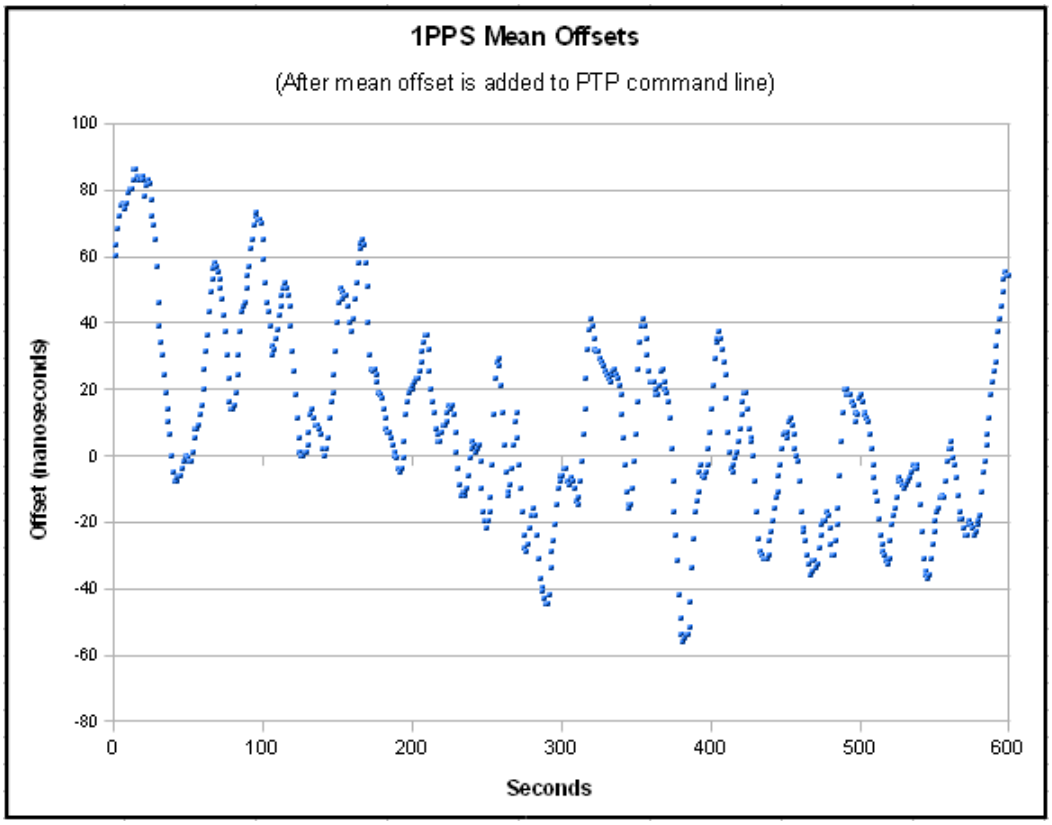

Figure 19: 1PPS Measurement with Offset

Solarflare sfptpd 1PPS I/O Specification

- 1PPS-input: SMA
  - Rising edge active, TTL into 50Ω
- 1PPS-output: SMA
  - Rising edge on-time, TTL into 50Ω
- Pulse Width
  - 200ms high, 800ms low

Chapter 10: Known Issues and Limitations

None.

Appendix A: Logging Options

The section explains and demonstrates the various data logging options available with sfptpd.

Event Logging

PTP events including startup events can be directed to the syslog or stderr by enabling the following option in the configuration file:

```
message-log [syslog | stderr]
```

Stats Logging

The following option is used to enable stats logging and display output on stdout or redirect output to a file:

```
stats-log [off | stdout | filename]
```

Enabling the stats-log option will produce output similar to the following:

```
2013-11-26 17:29:49.527405 [ptp-gm->phc0(eth2)], offset: 31.000, freq-adj: -
1237.563, in-sync: 1, one-way-delay: 42.000
```

Table 10: sfptpd Delay/Offset Data

Field	Description
`2013-11-26 17:29:49.527405`	Time of the log output.
`[ptp-gm->phc0(eth2)]`	A line of output generated for measurements between the external master clock and the Local Reference Clock.
`[phc0(eth2/eth3)->system]`	A line of output generated for measurements between the Local Reference Clock and the server system clock.
`offset (nanoseconds)`	The offset between the clocks.
`freq-adj`	The amount (PPB) by which the clock servo has adjusted the clock being disciplined.
`in-sync`	0 - offset is > 1 microsecond. 1 - offset is < 1 microsecond. The in-sync flag will change to a zero value if alarms conditions indicate a connection issue or loss of PTP messages from an external master clock.
`one-way-delay (nanoseconds)`	The network path delay between the slave server and remote master server.

If the stats-log output offset, one-way-delay or in-sync fields are coloured RED, it indicates an alarm condition in the PTP message sequence or PTP network - check the topology file for current alarm conditions. See The Toplogy File on page 34

PTP Packet Capture

Enabling the following option in the configuration file will display the contents of PTP packets received by the sfptpd process:

```
ptp-pkt-dump
```

This option produces extensive output for each received PTP packet, as the following example of a PTP SYNC message demonstrates, and **should only be used for debugging purposes**.

```
2012-12-20 18:45:29.496035 msgDebugHeader: messageType 0
2012-12-20 18:45:29.496035 msgDebugHeader: versionPTP 2
2012-12-20 18:45:29.496035 msgDebugHeader: messageLength 44
2012-12-20 18:45:29.496035 msgDebugHeader: domainNumber 0
2012-12-20 18:45:29.496035 msgDebugHeader: flags 02 00
2012-12-20 18:45:29.496035 msgDebugHeader: correctionfield 0
2012-12-20 18:45:29.496035 msgDebugHeader: sourcePortIdentity.clockIdentity
     000f:53ff:fe16:0474
2012-12-20 18:45:29.496035 msgDebugHeader: sourcePortIdentity.portNumber 1
2012-12-20 18:45:29.496035 msgDebugHeader: sequenceId 94
2012-12-20 18:45:29.496035 msgDebugHeader: controlField 0
2012-12-20 18:45:29.496035 msgDebugHeader: logMessageInterval 0
2012-12-20 18:45:29.496035 msgDebugSync: originTimestamp.seconds 1356029129
2012-12-20 18:45:29.496035 msgDebugSync: originTimestamp.nanoseconds 856792000
```

Note: This is different to using tcpdump which will capture packets received/sent at an interface.

Appendix B: 1PPS Statistical Data

1PPS statistical counters and error data is available from the following files:

```
/sys/class/net/eth<N>/device/pps_stats/<filename - see table>
```

Two sets of data are provided in the form of 1PPS offsets (min, max,mean and last) and 1PPS periods (min, max, mean and last). **All measurements are in nanoseconds**.

Table 11: PPS Statistics

File	Description
`pps_off_min`	Minimum offset value.
`pps_off_max`	Maximum offset value.
`pps_off_mean`	Average offset value observed over the last 8 values.
`pps_off_last`	Most recent offset value.
`pps_per_min`	Minimum 1PPS period.
`pps_per_max`	Maximum 1PPS period.
`pps_per_mean`	Average 1PPS period observed over the last 8 periods.
`pps_per_last`	Most recent 1PPS period.
`pps_oflow`	Too many 1PPS values received. Operation is suspended until the next sfptpd enable. This can occur when a cable is connected or as the result of a bad signal or noise on the 1PPS input. Re-start the sfptpd processes.
`pps_bad`	Very bad 1PPS period seen. The 1PPS period measured is too long to be a pulse per second i.e. period > 1 second. Check the 1PPS input is connected to a genuine 1PPS output.

Reset Statistics Counters

It is possible to reset 1PPS counters in the stats files by writing a '1' to the `ptp_stats` file relevant for the Solarflare interface.

```
echo 1 > /sys/class/net/eth<N>/device/ptp_stats
```

NOTE: root privileges are required to write to the ptp_stats files.

Appendix C: Transition Guide

This section is a transition guide to ease the task of porting Solarflare ptpd2 command line options to the Solarflare Enhanced PTP - sfptpd configuration file.

Configuration File Options

Table 12 identifies equivalent commands and lists new options available in sfptpd.

Table 12: Configuration Options

ptpd2	ptpd2 Description	sfptpd
`-a`	Specify clock servo proportional and integral attenuations.	no equivalent option.
`-b`	Bind PTP to network interface NAME.	Command line option: -i Config file option: **interface <interface-name>**. The interface the synchronization module should use.
`-B`	In time both, specify system time sync rate in seconds.	no equivalent option.
`-c`	Run in command line (non-daemon) mode.	no equivalent option.
`-d`	Display minimum stats (per received packet).	Config file option: **stats-log [off| stdout | filename]** Enable stats to stdout or redirect to file. By default stats logging is off.
`-D`	Display full stats (per received packet).	no equivalent option.
`-E`	Display full stats in csv format (per received packet).	no equivalent option.
`-f`	Redirect output to the specified file in csv format.	Config file option: **stats-log** <filename>
`-H`	Display help page.	Command line option: **-h**
`-i`	PTP domain. Default value is 0, Range 0:255.	Config file option: **ptp-domain <number>** Specify the PTP domain in the range 0:255, the default domain is 0.
`-j`	Allow clock stepping only once when the clock is initially updated at startup. The clock step can be of any magnitude and can be backwards or forwards. This should not be used with the -x option.	Config file option: **clock-control [slew-and-stop | step-at-startup | no-step | no adjust]** By default clocks are stepped and slewed as necessary.
`-l`	Specify inbound, outbound latency in nanoseconds (use this to compensate for asymmetry in the network).	Config file options: **ptp_tx_latency <number> ptp_rx_latency <number>** Specify inbound and outbound latency in nanoseconds.

Table 12: Configuration Options

ptpd2	ptpd2 Description	sfptpd
-L	Allow multiple instances of PTPd.	no equivalent command. Only a single instance of sfptpd can be running on the server.
-m	Specify the maximum number of foreign master records.	Config file option: **ptp-max-foreign-records <number>** Specify the number of foreign master records. The default is 16.
-M	This option disregards master to slave or slave to master delay measurements larger than the specified number of seconds. The master to slave delay is the difference between master sync transmit time and slave sync receive time. The slave to master delay is the difference between the slave Delay_Req transmit time and the master Delay_req receive time. The comparison is absolute so large negative delays are also discarded. VALUE can be specified as a decimal e.g. 0.123456789 Instances will prompt a syslog message e.g. "updateDelay aborted, s->m delay X.Y greater than administratively set maximum Z.A".	Config file option: **ptp-delay-discard-threshold <number>** Disregard delay measurements of more than <number> seconds.
-n	Specify announce interval in seconds.	Config file option: **ptp-announce-interval <number>** PTP announce message interval is 2^ <number> seconds
-o	Specify the current UTC offset. If an offset value is specified, this will set the UTC offset valid flag.	Config file option: **ptp-utc-offset <number>** The current UTC offset in seconds. If an offset value is specified this will set the UTC offset valid flag. Only applicable to master clock.
-p	Master mode only. Specify priority 1 attribute.	Config file option: **ptp-clock-priority1** <number>
-P	Display each received packet in detail.	Config file option: **ptp-pkt-dump** Display each received PTP packet in detail.
-q	Master mode only. Specify priority 2 attribute.	Config file option: **ptp-clock-priority2 <number>**
-r	Master mode only. Specify system clock accuracy.	Config file option: **ptp-clock-accuracy <number>**
-R	Record data about sync packets in a separate file.	no equivalent option.

Table 12: Configuration Options

ptpd2	ptpd2 Description	sfptpd
-s	Master mode only. Specify system clock class.	Config file option: **ptp-clock-class <number>**
-S	Don't send messages to syslog.	Config file option: **message-log [off | syslog | stderr]** If and where to log stats. By default stats logging is disabled. To view messages sent to syslog look in /var/log/messages
-t	Do not make changes to any clocks.	Config file option: **clock-control [slew-and-step | set-at-startup | no-step | no-adjust]** Determines how clocks are controlled. By default clocks are stepped and slewed as necessary.
-T	Set multicast time to live (TTL).	Config file option: **ptp-ttl <number>** The TTL value used in transmitted PTP packets. The default value is 1.
-v	Master mode only. Specify system clock Allen variance.	Config file option: **ptp-clock-allen-variance <number>.**
-w	Specify one way delay filter stiffness.	no equivalent option.
-x	Disable clock stepping. The adapter clock is set to the system clock at startup, but the system clock is unchanged. The SIGUSR1 signal can be used to step both adapter clock and system clocks to the 'offset from master' value during runtime. This should not be used with the -j option.	Config file option: **clock-control [slew-and-step | set-at-startup | no-step | no-adjust]** Determines how clocks are controlled. By default clocks are stepped and slewed as necessary.
-X	Selects which timer is used and controlled. both = NIC time is synchronized over the network via PTP and system time against NIC via local PTP (default). system = the host's system time. nic = the network interface. linux_hw = synchronize system time with Linux kernel assistance via net_tstamp API, uses NIC time stamping.	No equivalent option. Refer to configuration files for synchronization modes.
-y	Specify sync interval in seconds.	Config file option: **ptp-sync-pkt-interval <number>** Specify the rate as 2^ <number> seconds at which sync pkts are sent.

Table 12: Configuration Options

ptpd2	ptpd2 Description	sfptpd
`-Y`	Initial Delay_Req interval in seconds.	Config file option: **ptp-delayreq-interval <number>** The PTP Delay Request pkt interval is 2^ <number> seconds. If specified this value overrides the value communicated to the slave from the master clock.
`-Z`	Ignore Delay_Resp interval given by master.	no equivalent option.
`-g`	Run as slave only.	Config file option: **ptp-mode slave**
`-G`	Run as a master with NTP. Ensure the NTP service is running on the master.	Config file option: **ptp-mode master-ntp**
`-W`	Run as a master without NTP (reverts to slave mode when inactive). Ensure the NTP service is not running on the master.	Config file option: **ptp-mode master**
`SIGHUP`	Re-open statistics log specified with -f.	**SIGHUP** Rotate stats log (if logging to file).
`SIGINT`	Close file, remove lock file, clean exit.	no equivalent option.
`SIGKILL`	Unclean exit.	no equivalent option.
`SIGUSR1`	Manually step clock to current OFM value (overrides -x)	**SIGUSR1** Manually step clock to current OFM value.
`SIGUSR2`	Swap domain between current and current + 1	no equivalent option. see Config file option: **ptp-domain <number>** specify PTP domain in the range 0:255, default domain is 0.
	no equivalent option.	**persistent-clock-correction [off | on]** Specify whether to use the saved clock frequency corrections when disciplining clocks. On by default.
	no equivalent option.	**ptp-pps-log** Enable logging of PPS measurements.
	no equivalent option.	**ptp-network-mode [multicast | hybrid]** Default mode is multicast.
	no equivalent option.	**ptp-announce-timeout <number>** Announce receipt timeout as a number of Announce pkt intervals.