Solarflare Enhanced PTP User Guide

Copyright © 2014 SOLARFLARE Communications, Inc. All rights reserved.

The software and hardware as applicable (the “Product”) described in this document, and this document, are protected by copyright laws, patents and other intellectual property laws and international treaties. The Product described in this document is provided pursuant to a license agreement, evaluation agreement and/or non-disclosure agreement. The Product may be used only in accordance with the terms of such agreement. The software as applicable may be copied only in accordance with the terms of such agreement.

The furnishing of this document to you does not give you any rights or licenses, express or implied, by estoppel or otherwise, with respect to any such Product, or any copyrights, patents or other intellectual property rights covering such Product, and this document does not contain or represent any commitment of any kind on the part of SOLARFLARE Communications, Inc. or its affiliates.

The only warranties granted by SOLARFLARE Communications, Inc. or its affiliates in connection with the Product described in this document are those expressly set forth in the license agreement, evaluation agreement and/or non-disclosure agreement pursuant to which the Product is provided. EXCEPT AS EXPRESSLY SET FORTH IN SUCH AGREEMENT, NEITHER SOLARFLARE COMMUNICATIONS, INC. NOR ITS AFFILIATES MAKE ANY REPRESENTATIONS OR WARRANTIES OF ANY KIND (EXPRESS OR IMPLIED) REGARDING THE PRODUCT OR THIS DOCUMENTATION AND HEREBY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT, AND ANY WARRANTIES THAT MAY ARISE FROM COURSE OF DEALING, COURSE OF PERFORMANCE OR USAGE OF TRADE.

Unless otherwise expressly set forth in such agreement, to the extent allowed by applicable law (a) in no event shall SOLARFLARE Communications, Inc. or its affiliates have any liability under any legal theory for any loss of revenues or profits, loss of use or data, or business interruptions, or for any indirect, special, incidental or consequential damages, even if advised of the possibility of such damages; and (b) the total liability of SOLARFLARE Communications, Inc. or its affiliates arising from or relating to such agreement or the use of this document shall not exceed the amount received by SOLARFLARE Communications, Inc. or its affiliates for that copy of the Product or this document which is the subject of such liability.

The Product is not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

SF-109110-CD
Issue 3
# Table of Contents

**Chapter 1: What’s New** .................................................. 1

**Chapter 2: Introduction** .................................................. 3
  2.1 Purpose ................................................................. 3
  2.2 Definitions, Acronyms and Abbreviations ............................ 4

**Chapter 3: How PTP Works** .............................................. 5
  3.1 Message Sequence ..................................................... 5
  3.2 One-Way-Delay Interval .............................................. 6
  3.3 Announce Message .................................................... 6
  3.4 Solarflare sfptpd 2-stage synchronization .......................... 7
  3.5 Understanding the sfptpd startup sequence ......................... 7
  3.6 Understanding sfptpd output ....................................... 9

**Chapter 4: Overview** ..................................................... 12
  4.1 Solarflare PTP Network Adapters ................................. 12
  4.2 Timestamping Ports ................................................. 12
  4.3 Software support .................................................... 15
  4.4 Synchronization Model .............................................. 17
  4.5 Synchronization Modes ............................................. 17
  4.6 Transmission Modes ............................................... 21
  4.7 Saved Clock Frequency Correction Data ........................... 21
  4.8 Handling of Leap Seconds ......................................... 22
  4.9 System Time ......................................................... 22
  4.10 Loss of PTP Link Network Connection ............................ 22

**Chapter 5: Installation** .................................................. 23
  5.1 System Requirements ................................................. 23

**Chapter 6: Using sfptpd** ............................................... 27
  6.1 Download sfptpd ..................................................... 27
  6.2 Run sfptpd ............................................................ 28
  6.3 PTP over VLAN ....................................................... 29
  6.4 PTP over Bonded Interfaces ....................................... 29
  6.5 Hardware Timestamps ............................................... 30
  6.6 Hardware Timestamps Enable/Disable .............................. 31
  6.7 Hardware Timestamps (Kernel/Onload) ............................ 31
  6.8 PTP Management Messages ....................................... 31
  6.9 UTC Offset ........................................................... 32
  6.10 Access Control Lists ............................................... 33
  6.11 sfptpd in Operation ................................................. 34
  6.12 Accuracy under Network Load .................................... 36
Chapter 7: Configuration Files ................................................................. 37
7.1 Overview ......................................................................................... 37
7.2 Command Line options ................................................................. 39
7.3 Starting sfptpd with a config file ............................................... 39
7.4 Additional Options ........................................................................ 39
7.5 Synchronization Mode: PPS ......................................................... 39

Chapter 8: PTP State .............................................................................. 42
8.1 View Statistics Files ....................................................................... 42
8.2 PTP Nodes ...................................................................................... 42
8.3 PTP Interfaces ............................................................................... 42
8.4 The Topology File. .......................................................................... 42
8.5 Statistics Files. ............................................................................... 44

Chapter 9: Pulse Per Second (1PPS) ..................................................... 48
9.1 Asymmetric Networks ..................................................................... 48
9.2 1PPS Measurement Procedure ................................................... 49
9.3 1PPS in Practice ............................................................................. 51

Chapter 10: Known Issues and Limitations .......................................... 53

Appendix A: Logging Options .............................................................. 54

Appendix B: 1PPS Statistical Data ......................................................... 56
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 for received and transmitted PTP packets.
- 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.
- PTP Synchronization mode.
- PPS Synchronization mode.

This issue of the user guide supports sfptpd from version 2.2.2.60 which includes full support for the following Solarflare adapters:

- Solarflare Flareon™ Ultra SFN7002F, SFN7122F, SFN7322F and SFN7142Q.
- Solarflare SFN6322F adapter.
- Solarflare SFA6902F ApplicationOnload™ Engine.
- HP 570FLB FlexibleLOM and HP 570M Mezzanine adapters.

New Features

PPS Synchronization Mode

Version 2.2.2.60 sfptpd supports a new mode where the sfptpd slave can be synchronized to a PPS signal using NTP to provide a periodic time of day update.

For details of PPS synchronization mode and configuration requirements, please refer to PPS Mode on page 20 and Synchronization Mode: PPS on page 39.

PTP Management Messages

From version 2.2.2.60, sfptpd enables retrieval of PTP clock data by a remote management node with support for PTP management GET messages. Refer to PTP Management Messages on page 31 for the supported parameter list.
Access Control Lists

Access control lists are a means to restrict the set of network addresses a PTP slave will accept PTP timing and management messages from. Trusted IP addresses can be permitted access with all others denied. For further details and configuration examples, refer to Access Control Lists on page 33.

Forward Stepping Clock Control

A new clock_control step_forward option will ensure that clocks in a slave server are never stepped backward. See clock_control options in the sfptpd configuration files.

UTC offset Validation Handling

A new configuration file option controls how sfptpd handles UTC offset data transmitted from a PTP master clock. Refer to UTC Offset on page 32 for details.

Detect Transparent Clocks (topology file)

The PTP network topology file generated in the /var/lib/sfptpd directory has been enhanced to identify transparent clocks in the network path between slave and master PTP servers.

Report Active Network Clocks and PTP Domains

The ptp interfaces file and ptp nodes file are additional files created in the /var/lib/sfptpd directory to identify all active PTP clocks in the network and identify the timestamping capabilities of each interface in a PTP server. For examples see PTP Nodes and PTP Interfaces on page 42.

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 Topology File on page 42.

Configuration Files - Syntax Changes

While current versions of sfptpd support both dashes (-) and underscores (_) in configuration file parameters, future implementations will only support underscores.

Current versions support both:

ptp_delayreq_interval and ptp-delayreq-interval

Future implementations will support only the underscore syntax.
Chapter 2: Introduction

2.1 Purpose

This document describes Solarflare Enhanced PTP (sfptpd) support for Solarflare's 10GbE SFP+ and QSFP+ 40GbE 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 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.
- HP 570FLB FlexibleLOM and HP 570M Mezzanine adapters 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

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

![PTP Message Sequence Diagram]

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.

\[
\text{one\_way\_delay} = \frac{(T2-T1) + (T4-T3)}{2} \quad \text{offset} = \frac{(T2-T1) - (T4-T3)}{2}
\]

### 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 queuing these messages.

\[
\text{window} = (2^{\logMessagePeriod}) \times 2
\]

So a logMessagePeriod of 3:

\[
\text{window} = (2^3) \times 2 = 0-16 \text{ second window}
\]

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](image)

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 eth4 -fconfig/ptp_slave.cfg

1  2014-08-27 14:29:17.770703: info: Solarflare Enhanced PTP Daemon, version 2.2.2.60
9  [phc1(eth4/eth5)->system], offset: 2222141.625, freq-adj: -1090295.022, in-sync: 0
```
9  2014-08-27 14:29:19.120237 [phc1(eth4/eth5)->phc0(eth2)], offset: 219297.125, freq-adj: -93252.433, in-sync: 0


Table 1: sfptpd startup output

<table>
<thead>
<tr>
<th>Line #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Version of sfptpd running.</td>
</tr>
<tr>
<td>2</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>Create a PTP synchronization module.</td>
</tr>
<tr>
<td>4</td>
<td>The Local Reference Clock. From the SFN6322F 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.</td>
</tr>
<tr>
<td>5-6</td>
<td>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.</td>
</tr>
<tr>
<td>7</td>
<td>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.</td>
</tr>
<tr>
<td>8</td>
<td>Create an NTP synchronization module - always created, but only used in PPS synchronization mode.</td>
</tr>
</tbody>
</table>
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

\[
[ptp-gm->phc0(eth2)], \text{offset: 235.000, freq-adj: -1444.729, in-sync: 1, one-way-delay: 932.000} \quad \text{grandmaster-id: 000f:53ff:fe16:0474}
\]

### Table 1: sfptpd startup output

<table>
<thead>
<tr>
<th>Line #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>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.</td>
</tr>
<tr>
<td>10-11</td>
<td>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.</td>
</tr>
<tr>
<td>12</td>
<td>As the slave begins to synchronize the adapter clock with the external master clock, the offset (master-&gt;slave) and one-way-delay (slave-&gt;master) values are displayed as NANOSECONDS. The frequency correction value is the PPB value being used to discipline the clock to which the line of output refers.</td>
</tr>
</tbody>
</table>

### Table 2: sfptpd offset output

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ptp-gm-&gt;phc0(eth2)]</td>
<td>The values on this line show offset and one-way-delay between the master clock and the local identified clock.</td>
</tr>
<tr>
<td>freq-adj</td>
<td>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.</td>
</tr>
</tbody>
</table>
**Table 2: sfptpd offset output**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>offset</strong></td>
<td>The current offset (nanoseconds) between the adapter clock and the master clock.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><strong>in-sync</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>The in-sync flag will be 0 before the above condition is true.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Check the topology file for current alarms status.</td>
</tr>
<tr>
<td><strong>one-way-delay</strong></td>
<td>The current one-way-delay (nanoseconds) between master and slave servers.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Check the topology file for current alarms status.</td>
</tr>
<tr>
<td><strong>grandmaster-id</strong></td>
<td>Master clock UUID derived from its MAC address.</td>
</tr>
</tbody>
</table>
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.

NOTE: All files in the `/var/lib/sfptpd` directory are deleted when sfptpd starts. To preserve the stats_log files when logging to this directory the user should consider renaming the file to add a prefix e.g. `stats_log.txt` could be renamed `Aug03_stats_log.txt`. Any file called `stats-<anything>` will be deleted on startup.

When logging the stats_log to file, it is possible to override this using the `-v` option on the sfptpd command line to cause stats to display on stdout.
Chapter 4: Overview

4.1 Solarflare PTP Network Adapters

Solarflare time synchronization adapters 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 enables 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 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.

On HP branded 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+ and 40GbE QSFP+ Adapters

The SFN7000 series 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 any adapter port 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. SFN7142Q adapters will require the SOLR-PPS-DP40G bracket kit.

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

HP Branded Server Adapters

The HP570FLB and HP570M dual-port 10G Ethernet adapters are server adapters for the HP c-Class ProLiant Gen8 BladeSystem. These adapters are functionally equivalent to the Solarflare SFN7122F adapter and can support hardware timestamping on either port when the PTP/HW timestamping license is installed.

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; oscillator accuracy < 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 identifies the minimum software components required to support time synchronization server adapters.

To identify Solarflare adapters in a server, run the following command:

```
# lspci -vvv -d 1924: | grep Product
```

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>
```

Minimum driver and firmware versions required to support PTP are listed below.

• For SFN7142Q:
  - Linux net driver v4.1.0.6734
  - Controller firmware v4.1.1.1022

• For SFN7002F, SFN7122F, SFN7322F:
  - Linux net driver v4.0.2.6628
  - Controller firmware v4.0.1.6625

• For HP 570FLB and HP 570M:
  - Linux net driver v4.0.2.6628
  - Controller firmware v4.0.7.6711

• For SFN6322F:
  - Linux net driver v3.3.0.6246
  - Controller firmware v3.3.0.6247

Please refer to chapter 3 Installation on page 23 for instructions on updating the adapter firmware.
Network driver

The driver is distributed as a standalone RPM (source and DKMS) or with the OpenOnload/EnterpriseOnload distributions (listed below).

- OpenOnload from version 201310-u1.
- EnterpriseOnload from version 3.0.0.2.

See Installation on page 23 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 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 5 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.

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, PPS/NTP, 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  sync-mode pps
```
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.

![Synchronization Mode - PTP](image)

Figure 6: 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 7 below.
The system clock is used to set the LRC time once only when sfptpd starts and thereafter the adapter clock runs free.

Figure 7: 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 8: 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.
PPS Mode

In `sync_mode pps`, the slave server is running an NTP client allowing the system clock to be synchronized with an external NTP server. The slave server receives both NTP and 1PPS from the remote master clock. One adapter clock is selected as the LRC and NTP periodically provides the time of day to the slave server system clock and LRC. Thereafter sfptpd is accurately synchronizing the LRC from the 1PPS pulse and keeping all other clocks - including the system clock synchronized with the high precision oscillator on the adapter clock.

**Figure 9: Synchronization Mode - NTP - PPS**

When there is no Solarflare PTP adapter installed, NTP will provide time of day to the slave system clock.

In `sync_mode pps` the slave server is not sending or receiving PTP packets. Options within the `pps_slave.cfg` file control how sfptpd interacts with the local NTP daemon.

The `ntp_mode` option has three possible values:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>Interaction with the NTP daemon is not required.</td>
</tr>
<tr>
<td>read-only</td>
<td>sfptpd will query the NTP daemon, but will not modify NTP state.</td>
</tr>
<tr>
<td>control</td>
<td>sfptpd will query the NTP daemon and control NTP state.</td>
</tr>
</tbody>
</table>

With a default of 1 second, the `ntp_poll_interval` identifies how often (seconds) the local NTP daemon will be polled.

Refer to *Synchronization Mode: PPS on page 39* for details of NTP authentication configuration.
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 42 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

- SFN7000 series network server adapter.
- SFN5322F or SFN6322F adapter.
- Solarflare SFA6902F ApplicationOnload™ Engine.
- HP570FLB FlexibleLOM or HP570M Mezzanine adapter.
- 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 (see Software support on page 15) 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 update 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.
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 an 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 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.

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 SFN6322F 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 or 64bit binary:
SF-108909-LS is a 32bit binary tarball
SF-108910-LS is a 64bit binary tarball
SF-113121-LS is a 32bit binary RPM
SF-113122-LS is a 64bit binary RPM

Download the required sfptpd package from https://support.solarflare.com/.

Install tarball

1. Unpack the compressed file using the tar command e.g.

```bash
tar -zxvf SF-108910-LS-<version>_Solarflare_Enhanced_PTP_Daemon_sfptpd-_64_bit_binary_tarball.tgz
```

```
sfptpd-2.2.2.60.x86_64/
sfptpd-2.2.2.60.x86_64/sfptpd
sfptpd-2.2.2.60.x86_64/COPYRIGHT
sfptpd-2.2.2.60.x86_64/config/
sfptpd-2.2.2.60.x86_64/config/pps_slave.cfg
sfptpd-2.2.2.60.x86_64/config/default.cfg
sfptpd-2.2.2.60.x86_64/config/freerun.cfg
sfptpd-2.2.2.60.x86_64/config/ptp_master_ntp.cfg
sfptpd-2.2.2.60.x86_64/config/ptp_master.cfg
sfptpd-2.2.2.60.x86_64/config/ptp_slave.cfg
sfptpd-2.2.2.60.x86_64/init.d/
sfptpd-2.2.2.60.x86_64/init.d/sfptpd
```

An init.d script is supplied so that sfptpd can be stared using standard Linux service: start, stop, restart, status commands.
Install RPM

1. Unpack the RPM package:
   
   ```sh
tar -zxvf SF-113122-LS-<version>_Solarflare_Enhanced_PTP_Daemon_sfptpd-_64_bit_binary_RPM.tgz
```

2. Install the binary RPM:
   
   ```sh
   rpm -ivh sfptpd-<version>.x86_64.rpm
   Preparing... ###################################################
   [100%]
   1:sfptpd ################################################### [100%]
```

- The sfptpd executable will be installed into root’s path.
- Configuration files are located in the `/var/lib/sfptpd/examples` directory.
- All stats/state/topology files are located in the `/var/lib/sfptpd` directory.

6.2 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.
  
  ```sh
  ./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 or examples sub-directory
  
  ```sh
  ./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 or examples sub-directory
  
  ```sh
  ./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 or examples sub-directory

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

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

Run sfptpd as PTP Slave - PPS mode

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

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

6.3 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
```

6.4 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 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.

6.5 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
```

If hardware timestamping is required only for PTP packets, there is no need to enable this parameter.
6.6 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]

6.7 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). For more details of using hardware packet timestamps when using OpenOnload see the Onload User Guide (SF-104474-CD).

6.8 PTP Management Messages

PTP Management ‘GET’ messages are supported from sfptpd v2.2.1.58. These messages allow a network management node to retrieve PTP clock data from other PTP nodes in the network. Table 3 includes all supported managementId types.

<table>
<thead>
<tr>
<th>managementId</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGMT_ID_NULL</td>
<td>0x0000</td>
</tr>
<tr>
<td>MGMT_ID_CLOCK_DESCRIPTION</td>
<td>0x0001</td>
</tr>
<tr>
<td>MGMT_ID_USER_DESCRIPTION</td>
<td>0x0002</td>
</tr>
<tr>
<td>MGMT_ID_DEFAULT_DATA_SET</td>
<td>0x2000</td>
</tr>
<tr>
<td>MGMT_ID_CURRENT_DATA_SET</td>
<td>0x2001</td>
</tr>
<tr>
<td>MGMT_ID_PARENT_DATA_SET</td>
<td>0x2002</td>
</tr>
<tr>
<td>MGMT_ID_TIME_PROPERTIES_DATA_SET</td>
<td>0x2003</td>
</tr>
<tr>
<td>MGMT_ID_PORT_DATA_SET</td>
<td>0x2004</td>
</tr>
<tr>
<td>MGMT_ID_PRIORITY1</td>
<td>0x2005</td>
</tr>
<tr>
<td>MGMT_ID_PRIORITY2</td>
<td>0x2006</td>
</tr>
<tr>
<td>MGMT_ID_DOMAIN</td>
<td>0x2007</td>
</tr>
<tr>
<td>MGMT_ID_SLAVE_ONLY</td>
<td>0x2008</td>
</tr>
<tr>
<td>MGMT_ID_LOG_ANNOUNCE_INTERVAL</td>
<td>0x2009</td>
</tr>
</tbody>
</table>
Management messages are disabled by default. To enable sfptpd support for GET messages, set the following option in the PTP configuration file.

```
ptp_mgmt_msgs read-only
```


### 6.9 UTC Offset

The UTC offset (TAI - UTC) is conveyed to PTP slave clocks from a PTP master clock within the Sync or FollowUp messages. The UTC valid flag (UTCV) indicates whether the UTC offset time is considered valid by a master device and therefore should be used by a PTP slave device.

In the sfptpd configuration file, the `ptp_utc_handling` option identifies the action taken with the UTC offset value by sfptpd.

**Table 4: UTC Handling**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>If UTCV is set, use UTC, otherwise do not.</td>
</tr>
<tr>
<td>ignore</td>
<td>Ignore the UTC valid flag. Always apply the UTC offset.</td>
</tr>
<tr>
<td>prefer</td>
<td>Prefer master clocks that set the UTCV flag above those that do not.</td>
</tr>
<tr>
<td>require</td>
<td>Do not accept PTP messages from master clocks that do not set the UTCV flag.</td>
</tr>
</tbody>
</table>
6.10 Access Control Lists

Access control lists restrict the set of network addresses from which sfptpd will accept PTP timing and management messages.

From the PTP configuration file, access is controlled through separate permit and deny lists of IP addresses. A further configuration option dictates the order in which permit and deny lists are evaluated. Separate lists can be created for PTP timing messages and for PTP management messages.

```
ptp_timing_acl_permit 172.16.128.48/32 172.16.128.47/32
ptp_timing_acl_deny 172.16.128.0/21
```

In the above example, the first line identifies two IP addresses from which PTP timing messages will be accepted. The second line identifies that access from any IP address from subnet 172.16.128.0/21 will be denied.

```
ptp_timing_acl_order permit-deny
```

The access list order option dictates the order in which the permit and deny lists are evaluated. If a received PTP timing message source IP address matches an entry in the permit list, the message is accepted. If the address does not match an entry in the permit list, but is from subnet 172.16.128.0/21, it will be rejected. A PTP timing message from any other IP address will be accepted.

When a slave is required to accept PTP messages only from specified IP address(es), these should be identified using the permit list - without using a deny list and without using the acl_order option:

```
ptp_timing_acl_permit 172.16.128.48/24
```

In the above example PTP timing messages will only be accepted from a PTP clock source having the IP address 172.16.128.48/24. PTP timing messages from any other source will be ignored.

If access control lists are not specified or not enabled in the configuration file, sfptpd will accept PTP timing messages from any master clock identified by the Best Master Clock algorithm and will respond to PTP management messages from any IP address.
6.11 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](image1)

![Figure 12: Offset of the system clock to the adapter’s clock](image2)

![Figure 13: Offset of the server’s system clock to the PTP master](image3)
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**
6.12 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](image-url)
Chapter 7: Configuration Files

7.1 Overview

The sfptpd distribution unpacks default configuration files into the config or examples 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 or examples sub-directory.

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

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sync-mode</td>
<td>ptp</td>
<td>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.</td>
</tr>
<tr>
<td>ptp-network-mode</td>
<td>hybrid</td>
<td>Delay_REQ and Delay_RESP messages are sent as UDP unicast - all other PTP messages are multicast.</td>
</tr>
<tr>
<td>persistent-clock-correction</td>
<td>on</td>
<td>Periodically saved clock frequency corrections are used to discipline local clocks.</td>
</tr>
</tbody>
</table>
**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` or examples sub-directory.

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

Table 6 identifies the default PTP options configured when using the `ptp_slave.cfg` file.

### Table 6: Default PTP slave options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ptp-mode</td>
<td>slave</td>
<td>PTP slave clock mode.</td>
</tr>
<tr>
<td>ptp-stats</td>
<td>off</td>
<td>Controls the level of stats logging to stderr or file.</td>
</tr>
<tr>
<td>ptp-tx-latency</td>
<td>0</td>
<td>Outbound latency in nanoseconds.</td>
</tr>
<tr>
<td>ptp-rx-latency</td>
<td>0</td>
<td>Inbound latency in nanoseconds.</td>
</tr>
<tr>
<td>ptp-ttl</td>
<td>64</td>
<td>TTL value in transmitted PTP packets.</td>
</tr>
<tr>
<td>ptp-domain</td>
<td>0</td>
<td>PTP domain.</td>
</tr>
</tbody>
</table>

### Table 5: Default PTP master options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ptp-mode</td>
<td>master</td>
<td>PTP master clock mode.</td>
</tr>
<tr>
<td>ptp-stats</td>
<td>off</td>
<td>Controls the level of stats logging to stderr or file.</td>
</tr>
<tr>
<td>ptp-tx-latency</td>
<td>0</td>
<td>Outbound latency in nanoseconds.</td>
</tr>
<tr>
<td>ptp-rx-latency</td>
<td>0</td>
<td>Inbound latency in nanoseconds.</td>
</tr>
<tr>
<td>ptp-ttl</td>
<td>64</td>
<td>TTL value in transmitted PTP packets.</td>
</tr>
<tr>
<td>ptp-domain</td>
<td>0</td>
<td>PTP domain.</td>
</tr>
<tr>
<td>sync-mode</td>
<td>ptp</td>
<td>sfptpd synchronizes the LRC to a remote PTP master clock.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>With a Solarflare PTP adapter installed, the LRC disciplined by sfptpd is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the precision clock on the adapter. sfptpd uses a second clock servo to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>synchronize the system clock and a clock servo for each additional Solarflare</td>
</tr>
<tr>
<td></td>
<td></td>
<td>adapter clock in the server. When no Solarflare PTP adapter is present, the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LRC is the system clock.</td>
</tr>
<tr>
<td>message-log</td>
<td>stderr</td>
<td>Direct PTP event messages to stderr.</td>
</tr>
<tr>
<td>ptp-network-mode</td>
<td>hybrid</td>
<td>Delay_Req and Delay_Resp messages are sent as UDP unicast - all other PTP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>messages are multicast.</td>
</tr>
<tr>
<td>persistent-clock-correction</td>
<td>on</td>
<td>Periodically saved clock frequency corrections are used to discipline local clocks.</td>
</tr>
<tr>
<td>ptp-mode</td>
<td>slave</td>
<td>PTP slave clock mode.</td>
</tr>
<tr>
<td>ptp-stats</td>
<td>off</td>
<td>Controls the level of stats logging to stderr or file.</td>
</tr>
<tr>
<td>ptp-tx-latency</td>
<td>0</td>
<td>Outbound latency in nanoseconds.</td>
</tr>
<tr>
<td>ptp-rx-latency</td>
<td>0</td>
<td>Inbound latency in nanoseconds.</td>
</tr>
</tbody>
</table>
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.

7.5 Synchronization Mode: PPS

Using sync_mode pps the sfptpd slave will receive NTP and PPS from a remote master clock. The slave runs an NTP client which sfptpd can both read and control.

In the /config/pps_slave.cfg file there are several configurable parameters:

- **pps_delay**
  
  This option sets the 1PPS pulse propagation delay (seconds). The propagation delay should be estimated for the length of the PPS cable between PPS source and PPS input on the adapter.

- **ntp_mode read-only**
  
  In read-only mode sfptpd will query the local NTP client to set the current time-of-day on the slave system clock.

  The following options can be enabled in the /etc/ntp.conf file to control NTP interaction with the system clock:
disable ntp

- prevent NTP from adjusting the system clock.

disable kernel

- prevent NTP from adjusting its own internal clock.

• ntp_mode control

In control mode sfptpd has full control of the local NTP daemon and can modify it’s state as required, for example to prevent NTP from disciplining the system clock. NTP daemon state changes are handled internally by sfptpd and are not configurable by the user.

• ntp_poll_interval

The poll_interval determines how often sfptpd will query the local NTP client.

NTP Authentication.

Before sfptpd can query the local NTP daemon, it is necessary to setup symmetric key authentication parameters in the NTP daemon configuration files and in the PTP configuration file.

1 The following is an extract from a typical /etc/ntp.conf file:

```
# Key file containing the keys and key identifiers used when operating # with symmetric key cryptography.
#keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
#controlkey 8
```

2 Uncomment the keys, trustedkey and requestkey lines and replace/add an integer value:

```
keys /etc/ntp/keys
trustedkey 15
requestkey 15
```

3 Append the following line to the file /etc/ntp/keys

```
15 M my authentication string
```
4 Restart the ntpd service

    # service ntpd restart

5 Edit the /config/pps_slave.cfg file to match the values specified for NTP:

    ntp_mode control
    ntp_key 15 "my authentication string"

6 Start the sfptpd daemon.
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 PTP Nodes

The PTP nodes file identifies all PTP clocks present in the network.

# cat ptp-nodes
<table>
<thead>
<tr>
<th>state</th>
<th>clock-id</th>
<th>port-number</th>
<th>domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>000f:53ff:fe16:0474</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

8.3 PTP Interfaces

The PTP interfaces file identifies all adapters in the PTP slave server and identifies the timestamping capabilities. In the following example, Solarflare adapter interfaces as identified as being hardware timestamping capable.

# cat interfaces
<table>
<thead>
<tr>
<th>interface</th>
<th>ptp-caps</th>
<th>pkt-timestamping-caps</th>
<th>product-name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eth0</td>
<td>-</td>
<td>-</td>
<td>Broadcom NetXtreme II Ethernet Controller</td>
</tr>
<tr>
<td>eth1</td>
<td>-</td>
<td>-</td>
<td>Broadcom NetXtreme II Ethernet Controller</td>
</tr>
<tr>
<td>eth2</td>
<td>hw</td>
<td>-</td>
<td>Solarflare SFN5322F SFP+ Time Stamping Server Adapter</td>
</tr>
<tr>
<td>eth3</td>
<td>-</td>
<td>-</td>
<td>Solarflare SFN5322F SFP+ Time Stamping Server Adapter</td>
</tr>
<tr>
<td>eth4</td>
<td>hw</td>
<td>hw</td>
<td>Solarflare SFN7122F SFP+ Server Adapter</td>
</tr>
<tr>
<td>eth5</td>
<td>hw</td>
<td>hw</td>
<td>Solarflare SFN7122F SFP+ Server Adapter</td>
</tr>
</tbody>
</table>

8.4 The Topology 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 will also 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 an example of a topology file showing alarm states.

```plaintext
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.
8.5 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 7 lists statistics files created by `sfptpd`.

<table>
<thead>
<tr>
<th>File name</th>
<th>Persistent</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>freq-correction-&lt;UUID&gt;</code></td>
<td>YES</td>
<td>Contains the frequency correction value used to discipline the clock.</td>
</tr>
<tr>
<td><code>freq-correction-system</code></td>
<td>YES</td>
<td>Contains the frequency correction value used to discipline the server system clock.</td>
</tr>
<tr>
<td><code>state-system</code></td>
<td>NO</td>
<td>The contents of this file depend on the current PTP mode. In slave mode this file contains historical offset and status data for the slave server.</td>
</tr>
<tr>
<td><code>stats-system</code></td>
<td>NO</td>
<td>Contains accumulated PTP performance data for the server including counts of the PTP message types sent and received.</td>
</tr>
<tr>
<td><code>state-&lt;UUID&gt;</code></td>
<td>NO</td>
<td>Contains status information relevant to the clock identified by the UUID identifier.</td>
</tr>
<tr>
<td><code>stats-&lt;UUID&gt;</code></td>
<td>NO</td>
<td>Contains accumulated synchronization data for the clock identified by the UUID identifier.</td>
</tr>
<tr>
<td><code>topology</code></td>
<td>NO</td>
<td>See above for an in depth description of this file.</td>
</tr>
</tbody>
</table>
The following tables identify and describe sfptpd stats files:

### Table 8: FILE: freq-correction-<UUID>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>freq-correction-&lt;UUID&gt;</td>
<td>This identifier is constructed from the hardware address of the clock port.</td>
</tr>
<tr>
<td>value</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

### Table 9: FILE: state-<UUID>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clock-name</td>
<td>identify the clock using this clock-id.</td>
</tr>
<tr>
<td>clock-id</td>
<td>the clock UUID.</td>
</tr>
<tr>
<td>state</td>
<td>ptp_slave</td>
</tr>
<tr>
<td>interface</td>
<td>identifies the PTP clock interface.</td>
</tr>
<tr>
<td>timestamping</td>
<td>current timestamping mode.</td>
</tr>
<tr>
<td>offset-from -master</td>
<td>Offset (nanoseconds) of LRC from the master clock.</td>
</tr>
<tr>
<td>one-way-delay</td>
<td>One-way-delay (nanoseconds) between LRC and master clock.</td>
</tr>
<tr>
<td>freq-adjustment-ppb</td>
<td>Current frequency correction value used to discipline this clock.</td>
</tr>
<tr>
<td>observed-drift</td>
<td>Drift in nanoseconds of slave LRC to master clock.</td>
</tr>
</tbody>
</table>
### Table 9: FILE: state-<UUID>

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

### Table 10: FILE: stats-<UUID>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NOTE:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td>offset-from-master</td>
<td>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.</td>
</tr>
<tr>
<td>freq-adjustment-ppb</td>
<td></td>
</tr>
<tr>
<td>one-way-delay</td>
<td></td>
</tr>
</tbody>
</table>
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 or 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The stats file retains accumulated counts of each PTP message type</td>
<td>The stats file retains accumulated counts of each PTP message type sent or received from the server.</td>
</tr>
<tr>
<td>The statistics file also records the number of PTP packets sent or</td>
<td>The statistics file also records the number of PTP packets sent or received without being hardware timestamped.</td>
</tr>
<tr>
<td>The number of samples taken during each period is recorded as is the</td>
<td>The number of samples taken during each period is recorded as is the start/end time of each sample period.</td>
</tr>
<tr>
<td>start/end time of each sample period.</td>
<td></td>
</tr>
</tbody>
</table>
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](image)

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 support 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 51.

### 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 **pps_off_mean** file to identify the mean offset value. Refer to **Appendix B: 1PPS Statistical Data** on page 56 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:

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).

Figure 17: 1PPS - Example
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 $\text{pps\_off\_mean: } -3450$. Refer to Appendix B: 1PPS Statistical Data on page 56 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.

```plaintext
ptp_rx_latency  6900  ptp_tx_latency  0
```

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

---

**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

---

Figure 19: 1PPS Measurement with Offset
Chapter 10: Known Issues and Limitations

Firmware Upgrade.

The sfptpd daemon must be terminated before upgrading the adapter firmware. Following firmware upgrade the adapter driver should be reloaded.

If onload is installed:

```bash
# onload_tool reload
```

If onload is not installed:

```bash
# modprobe -r sfc
# modprobe sfc
```

PTP and SolarCapture

When using Solarflare’s SolarCapture to capture packets from an interface also being used to send/receive PTP messages, PTP hybrid mode will not function correctly when SolarCapture consumes the ARP response messages. This prevents the unicast Delay_Request messages being sent from the PTP slave. sfptpd in multicast mode is not affected and users are advised to select multicast mode in the PTP configuration file.

```bash
ptp_network_mode multicast
```

Refer to the SolarCapture User Guide configuration options when using SolarCapture and sfptpd on the same server.

LACP Bonding

Using the Linux bonding driver, sfptpd supports active/passive bonding. PTP over 802.3.ad LACP bonded interfaces is not supported.
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>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-11-26 17:29:49.527405</td>
<td>Time of the log output.</td>
</tr>
<tr>
<td>[ptp-gm-&gt;phc0(eth2)]</td>
<td>A line of output generated for measurements between the external master clock and the Local Reference Clock.</td>
</tr>
<tr>
<td>[phc0(eth2/eth3)-&gt;system]</td>
<td>A line of output generated for measurements between the Local Reference Clock and the server system clock.</td>
</tr>
<tr>
<td>offset (nanoseconds)</td>
<td>The offset between the clocks.</td>
</tr>
<tr>
<td>freq-adj</td>
<td>The amount (PPB) by which the clock servo has adjusted the clock being disciplined.</td>
</tr>
<tr>
<td>in-sync</td>
<td>0 - offset is &gt; 1 microsecond. 1 - offset is &lt; 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.</td>
</tr>
<tr>
<td>one-way-delay (nanoseconds)</td>
<td>The network path delay between the slave server and remote master server.</td>
</tr>
</tbody>
</table>
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 Topology File on page 42

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 12: PPS Statistics

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

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.