BCM5750X

Precision Time Protocol

Application Note

Table of Contents

1 Precision Time Protocol Introduction

Precision Time Protocol (PTP) is a protocol that is used to synchronize the clocks of devices on a network to a high level of accuracy. It is based on the Internet Protocol (IP) and is designed to synchronize the time of day across multiple devices that are connected to a network.

PTP operates by using a source clock that sends out time synchronization messages to other devices on the network, which are known as sink clocks. The sink clocks then adjust their own internal clocks based on the messages received from the source clock. PTP achieves time synchronization by timestamping the messages to determine the network delay between the sending and receiving nodes.

PTP is used in a variety of applications where accurate time synchronization is important, such as in financial systems, power grids, and telecommunications networks. It is also used in scientific and industrial applications where precise time measurement is required.

2 PTP Specification

The main standard that specifies the Precision Time Protocol (PTP) is IEEE 1588.

The IEEE 1588 standard defines the protocol and the messages used by PTP to synchronize the clocks of devices on a network. It also specifies the hardware and software requirements for implementing PTP as well as the procedures and algorithms used to ensure accurate time synchronization.

BCM5750X network controllers are compliant with the IEEE1588v2-2019/08 standard.

2.1 PTP Messages

In the PTP, several types of messages are used to synchronize the clocks of devices on a network. These messages are sent between the source clock and the sink clocks on the network.

1. Event messages: Time-critical messages with an accurate timestamp that is generated at both the transmit time and receive time.

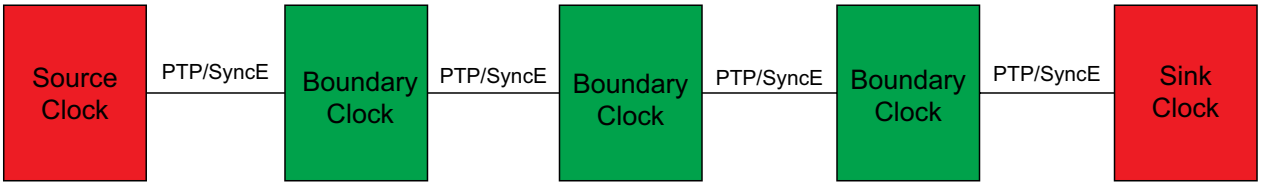
   Example event messages:
   - Sync message: This message is sent by the source clock to the sink clocks and contains the current time as measured by the source clock.
   - Follow-up message: This message is sent by the source clock in response to a sync message received from a sink clock. It contains additional information about the timing of the sync message, including the time it was sent and received.
   - Delay request message: This message is sent by a sink clock to the source clock and is used to request a delay measurement.
   - Delay response message: This message is sent by the source clock in response to a delay request message and contains the delay measurement between the source clock and the sink clock.
2. Announce message: This message is sent by the source clock and contains information about the current state of the PTP network, including the identity of the source clock and the sink clocks, as well as the current time.
3. Management message: This message is used by network management to monitor, configure, and maintain a PTP system
4. Signaling message: This message is used to initiate or terminate PTP communication between the source clock and the sink clocks. It can also be used to request a change in the state of the PTP network, such as switching to a new source clock.

2.2 PTP System Configurations

There are three primary PTP system configurations:

- Ordinary clock – A system with one connection to the 1588v2 network. That connection/port may be a source (supplies Time of day) or sink (must be supplied time of day)
- Boundary clock: system with multiple connections - one source port and one or more sink ports
- Transparent clock – Modifies the PTP messages as they pass through, but can correct network delays to improve the accuracy of the time distribution. This has two sub-modes: End-to-End mode and Peer-to-Peer mode
  - End-to-End mode – In this mode, the delay is measured between the sink and the source. The source and sink send IEEE 1588 messages called DELAY REQUEST and DELAY RESPONSE between the two, allowing the delay to be measured.
  - Peer-to-Peer mode – In this mode, the delay is measured at each network element between its input port and the device attached to the other end of the wire of this input port (the peer device). As the source sends its view of time (using SYNC messages) towards sink(s), each network element along the way receives the SYNC message and adds a correction to the SYNC message. The correction includes the measured wire delay of the input port the SYNC message was received on.

Figure 1: PTP System Configurations



2.3 PTP Profiles

A PTP profile is a set of predefined configuration options for a PTP system. The PTP standard, IEEE 1588, defines several profiles that are intended for different types of applications and networks. These profiles specify different options for message timing, delay measurement, and other PTP parameters.

The main profiles defined in IEEE 1588 are:

- Default – The default profile is intended for general use and provides a balance between accuracy and network overhead.
- Power – The power profile is intended for use in power distribution systems and has a lower accuracy than the default profile.
- Telecommunication – The telecommunication profile is intended for use in telecommunication networks and has a higher accuracy than the default profile.
- High-accuracy – The high-accuracy profile is intended for use in high-accuracy applications and has the highest accuracy of all the profiles. For example, the G.8257.1 profile is considered a high-accuracy profile and is widely used in different types of networks such as mobile networks, packet-based networks, and also some specific applications such as power distribution systems and telecommunication networks.

Each profile specifies different options for message timing, delay measurement, and other PTP parameters that are tailored to the needs of a specific application or network.

The BCM5750X controller supports G.8275.1, G.8275.2, and 802.1AS profiles.

3 PTP Delay Measurements

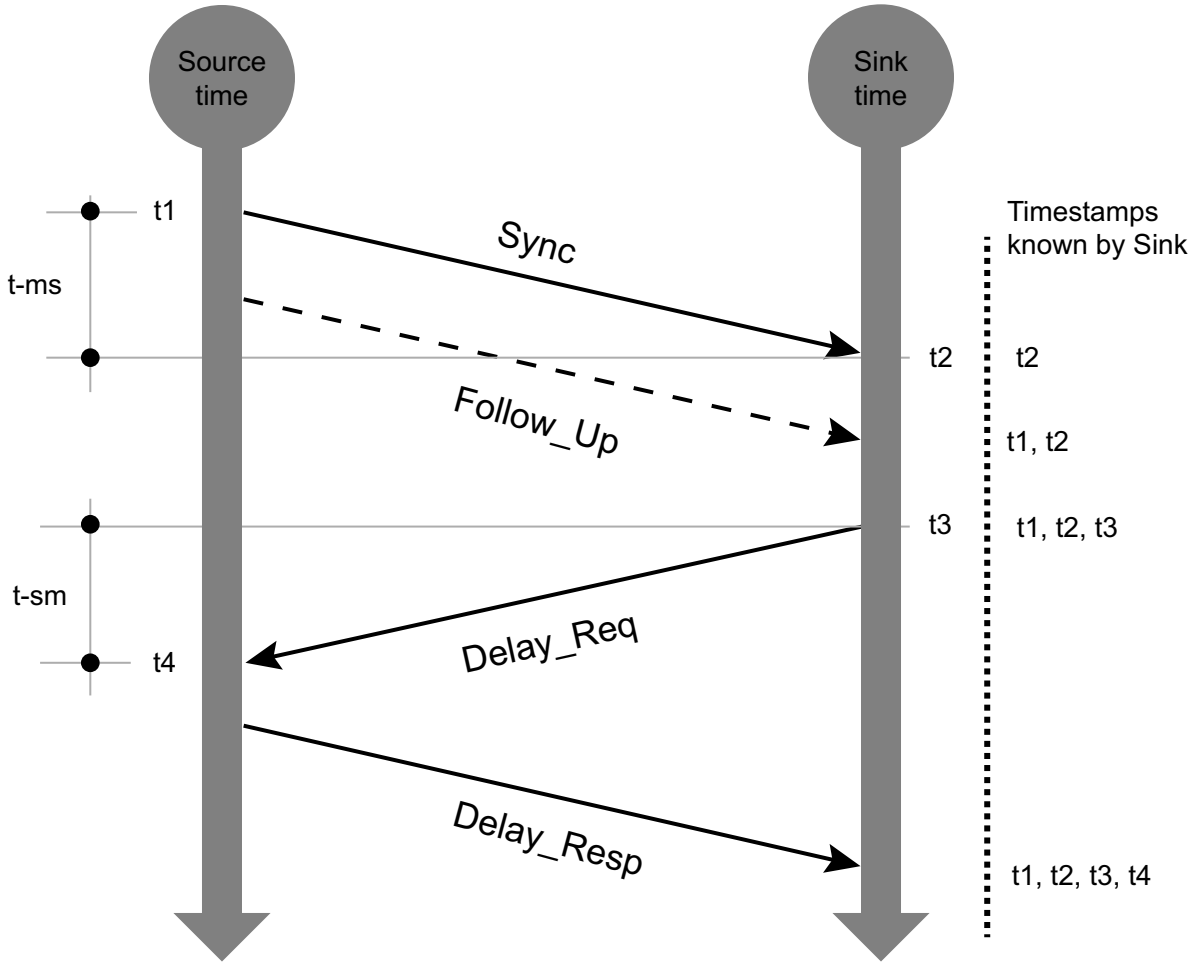
In PTP, a sink clock determines the time by receiving time synchronization messages from the source clock and adjusting its own internal clock based on these messages.

The BCM5750X controller supports two-step synchronization for synchronizing the time of day between the source clock and the sink clocks. In the two-step time synchronization process, the source clock sends a sync message to the sink clocks, and the sink clocks send back a follow-up message containing additional information about the timing of the sync message. The source clock then uses this information to compute the delay between the source clock and the sink clocks and sends a delay response message back to the sink clocks. The sink clocks use this information to further adjust their internal clocks and improve the accuracy of the time synchronization. Figure 2 describes how the two-step synchronization operates.

The general formula for two-step synchronization is as follows:

$$\text{Delay} = (t2 - t1) - \tfrac{1}{2}(t_{ms} + t_{sm})$$
$$= (t2 - t1) - \tfrac{1}{2}\{(t4 - t1) - (t3 - t2)\}$$

Figure 2: Two-Step Synchronization



4 Installing PTP

In Linux, PTP support in the software stack is provided by a package known as linuxptp, a PTPv2 implementation according to the IEEE standard 1588 for Linux. The linuxptp package includes the ptp4l and phc2sys programs for clock synchronization. The ptp4l program implements the PTP boundary clock and ordinary clock. With hardware time stamping, it is used to synchronize the PTP hardware clock to the source clock. With software time stamping, it synchronizes the system clock to the source clock. The phc2sys program is needed only with hardware time stamping for synchronizing the system clock to the PTP hardware clock on the network interface card (NIC).

To install linuxptp in Centos, use `yum install linuxptp`. To install linuxptp in Ubuntu, use `apt-get install linuxptp`. This installs ptp4l and phc2sys.

4.1 Checking for Driver and Hardware Support

To use PTP, the kernel network driver for the intended interface has to support either software or hardware time stamping capabilities. In addition to hardware time stamping support being present in the driver, the Ethernet adapter, in this case the BCM5750X, must also be capable of supporting this functionality in the physical hardware. The best way to verify the time stamping capabilities of a particular driver and Ethernet adapter is to use the ethtool utility to query the interface as follows:

```
[root@localhost ~]# ethtool -T ens106f0
Time stamping parameters for ens106f0:
Capabilities:
        hardware-transmit     (SOF_TIMESTAMPING_TX_HARDWARE)
        software-transmit     (SOF_TIMESTAMPING_TX_SOFTWARE)
        hardware-receive      (SOF_TIMESTAMPING_RX_HARDWARE)
        software-receive      (SOF_TIMESTAMPING_RX_SOFTWARE)
        software-system-clock (SOF_TIMESTAMPING_SOFTWARE)
        hardware-raw-clock    (SOF_TIMESTAMPING_RAW_HARDWARE)
PTP Hardware Clock: 3
Hardware Transmit Timestamp Modes:
        off                   (HWTSTAMP_TX_OFF)
        on                    (HWTSTAMP_TX_ON)
Hardware Receive Filter Modes:
        none                  (HWTSTAMP_FILTER_NONE)
        all                   (HWTSTAMP_FILTER_ALL)
        ptpv2-l4-event        (HWTSTAMP_FILTER_PTP_V2_L4_EVENT)
        ptpv2-l2-event        (HWTSTAMP_FILTER_PTP_V2_L2_EVENT)
```

For software time stamping support, the parameters list should include the following:

```
SOF_TIMESTAMPING_SOFTWARE
SOF_TIMESTAMPING_TX_SOFTWARE
SOF_TIMESTAMPING_RX_SOFTWARE
```

For hardware time stamping support, the parameters list should include the following:

```
SOF_TIMESTAMPING_RAW_HARDWARE
SOF_TIMESTAMPING_TX_HARDWARE
SOF_TIMESTAMPING_RX_HARDWARE
```

As seen from the previous figure, the BCM5750X Ethernet adapter supports hardware time stamping and is capable of providing hardware timestamps for both transmit and receive packets with different filter modes.

5 Configuring PTP

In this section two BCM5750X Ethernet adapters are configured with PTP, connected back-to-back, with one acting as source and the other as sink.

For ptp4l, the command line options and other options, which cannot be set on the command line, can be set in an optional configuration file. The configuration file must be specified at runtime with the -f option.

The configuration files for the profiles discussed in PTP Profiles can be found under directory `/usr/share/doc/linuxptp/configs/` when linuxptp is installed using yum.

In the ptp4l conf file, add the parameter `tx_timestamp_timeout` with a value of 50 to allow slightly more time for the driver's work thread to fetch the TX timestamp from the chip.

On the source, run the below command:

```
ptp4l -i <NIC interface name> -m -4 -f /etc/ptp4l.cfg
```

On the sink, run the below command:

```
ptp4l -i <NIC interface name> -m -s -4 -f /etc/ptp4l.cfg
```

The client will be in sync with the source as seen in the following figure where the sink is in S2 servo state with minimal changes to the freq and delay offsets.

Refer to Appendix A, ptp4l Configuration File for the contents of the ptp4l config file used in the previous instructions.

```
[root@dhcp-10-193-55-197 ~]# ptp4l -i ens106f0 -m -s -4 -f /etc/ptp4l.cfg
ptp4l[179585.569]: selected /dev/ptp2 as PTP clock
ptp4l[179585.612]: port 1: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[179585.613]: port 0: INITIALIZING to LISTENING on INIT_COMPLETE
ptp4l[179590.324]: port 1: new foreign master b02628.fffe.f55d1c-1
ptp4l[179592.655]: selected local clock bc97e1.fffe.ab6d78 as best master
ptp4l[179594.324]: selected best master clock b02628.fffe.f55d1c
ptp4l[179594.324]: port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[179596.324]: master offset   -1989300 s0 freq  -11608 path delay        59
ptp4l[179597.324]: master offset   -1989467 s1 freq  -11775 path delay        59
ptp4l[179598.324]: master offset       -829 s2 freq  -12604 path delay        59
ptp4l[179598.324]: port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[179599.324]: master offset          3 s2 freq  -12021 path delay        59
ptp4l[179600.324]: master offset        290 s2 freq  -11733 path delay        15
ptp4l[179601.324]: master offset        249 s2 freq  -11687 path delay        15
ptp4l[179602.324]: master offset        157 s2 freq  -11704 path delay        19
ptp4l[179603.324]: master offset         86 s2 freq  -11728 path delay        19
ptp4l[179604.324]: master offset         38 s2 freq  -11750 path delay        19
ptp4l[179605.324]: master offset         17 s2 freq  -11760 path delay        17
ptp4l[179606.324]: master offset          2 s2 freq  -11770 path delay        16
ptp4l[179607.324]: master offset         -2 s2 freq  -11773 path delay        14
ptp4l[179608.324]: master offset         -2 s2 freq  -11774 path delay        13
ptp4l[179609.325]: master offset         -2 s2 freq  -11774 path delay        13
ptp4l[179610.325]: master offset         -1 s2 freq  -11774 path delay        13
ptp4l[179611.324]: master offset          0 s2 freq  -11773 path delay        11
ptp4l[179612.325]: master offset          3 s2 freq  -11770 path delay         9
ptp4l[179613.325]: master offset         -2 s2 freq  -11774 path delay        10
ptp4l[179614.325]: master offset         -1 s2 freq  -11774 path delay        10
ptp4l[179615.326]: master offset          2 s2 freq  -11771 path delay         8
ptp4l[179616.325]: master offset          1 s2 freq  -11772 path delay         8
ptp4l[179617.325]: master offset          1 s2 freq  -11771 path delay         8
```

5.1 Synchronize System Times of Source and Sink

Start phc2sys to synchronize the system times between the source and the sink.

On source, run the following command:

```
phc2sys -s CLOCK_REALTIME -c <NIC interface name> -m -w
```

On sink, run the following command:

```
phc2sys -s <NIC interface name> -c CLOCK_REALTIME -m -w
```

After several seconds, the system clocks are synchronized.

Ensure that phc2sys on the source and client are not reporting large maxed-out frequency corrections. The following figure shows an acceptable phc offset with respect to CLOCK_REALTIME.

```
phc2sys[179711.498]: CLOCK_REALTIME phc offset    813289 s0 freq   +5251 delay    763
phc2sys[179712.499]: CLOCK_REALTIME phc offset    503961 s1 freq -303928 delay    732
phc2sys[179713.499]: CLOCK_REALTIME phc offset    142573 s2 freq -161355 delay    729
phc2sys[179714.500]: CLOCK_REALTIME phc offset    240842 s2 freq  -20314 delay    758
phc2sys[179715.500]: CLOCK_REALTIME phc offset    248632 s2 freq  +59728 delay    769
phc2sys[179716.500]: CLOCK_REALTIME phc offset    189917 s2 freq  +75603 delay    770
phc2sys[179717.500]: CLOCK_REALTIME phc offset    112714 s2 freq  +55375 delay    732
phc2sys[179718.500]: CLOCK_REALTIME phc offset     54269 s2 freq  +30744 delay    768
phc2sys[179719.500]: CLOCK_REALTIME phc offset     24912 s2 freq  +17668 delay    769
phc2sys[179720.501]: CLOCK_REALTIME phc offset     13923 s2 freq  +14153 delay    760
phc2sys[179721.501]: CLOCK_REALTIME phc offset      7117 s2 freq  +11523 delay    737
phc2sys[179722.501]: CLOCK_REALTIME phc offset      2714 s2 freq   +9256 delay    760
phc2sys[179723.501]: CLOCK_REALTIME phc offset       -32 s2 freq   +7324 delay    766
phc2sys[179724.501]: CLOCK_REALTIME phc offset     -1668 s2 freq   +5678 delay    763
phc2sys[179725.502]: CLOCK_REALTIME phc offset     -1745 s2 freq   +5101 delay    763
phc2sys[179726.502]: CLOCK_REALTIME phc offset     -1739 s2 freq   +4583 delay    759
phc2sys[179727.502]: CLOCK_REALTIME phc offset      -925 s2 freq   +4876 delay    759
phc2sys[179728.502]: CLOCK_REALTIME phc offset      -305 s2 freq   +5218 delay    763
phc2sys[179729.502]: CLOCK_REALTIME phc offset        -9 s2 freq   +5423 delay    763
phc2sys[179730.502]: CLOCK_REALTIME phc offset        70 s2 freq   +5499 delay    758
phc2sys[179731.503]: CLOCK_REALTIME phc offset        53 s2 freq   +5503 delay    763
phc2sys[179732.503]: CLOCK_REALTIME phc offset        81 s2 freq   +5547 delay    766
phc2sys[179733.503]: CLOCK_REALTIME phc offset        29 s2 freq   +5519 delay    766
phc2sys[179734.503]: CLOCK_REALTIME phc offset       -45 s2 freq   +5454 delay    759
phc2sys[179735.503]: CLOCK_REALTIME phc offset       -44 s2 freq   +5441 delay    763
phc2sys[179736.504]: CLOCK_REALTIME phc offset       -20 s2 freq   +5452 delay    763
phc2sys[179737.504]: CLOCK_REALTIME phc offset        -8 s2 freq   +5458 delay    767
phc2sys[179738.504]: CLOCK_REALTIME phc offset       -31 s2 freq   +5433 delay    772
phc2sys[179739.504]: CLOCK_REALTIME phc offset       -39 s2 freq   +5415 delay    768
phc2sys[179740.504]: CLOCK_REALTIME phc offset       -22 s2 freq   +5421 delay    763
phc2sys[179741.504]: CLOCK_REALTIME phc offset       -14 s2 freq   +5422 delay    768
phc2sys[179742.505]: CLOCK_REALTIME phc offset         2 s2 freq   +5434 delay    758
phc2sys[179743.505]: CLOCK_REALTIME phc offset         1 s2 freq   +5433 delay    760
phc2sys[179744.505]: CLOCK_REALTIME phc offset        14 s2 freq   +5447 delay    768
phc2sys[179745.505]: CLOCK_REALTIME phc offset         7 s2 freq   +5444 delay    769
phc2sys[179746.505]: CLOCK_REALTIME phc offset         0 s2 freq   +5439 delay    766
phc2sys[179747.505]: CLOCK_REALTIME phc offset        -2 s2 freq   +5437 delay    768
```

All clocks are now synchronized.

For best results use the following order:

1. Start phc2sys on both source and sink (they will not start and will show a message waiting for ptp4l).
2. Start ptp4l on source and then on sink.
3. The source is in Grandmaster mode. phc2sys on the source starts, sink ptp4l goes into s2 state, and finally phc2sys on the sink starts.

This ensures that when ptp4l starts, if there are large adjustments to be made, it is taken care of in s0/s1 state before entering s2 state. Once ptp4l is in s2 state, only small adjustments can be made.

Appendix A: ptp4l Configuration File

The following is the ptp4l config file that is used in the above example.

```
cat /etc/ptp4l.cfg
[global]
tx_timestamp_timeout 50
```

Revision History

5750X-AN300; February 16, 2023

Initial release.