## Application Note:

# Precision Time Protocol IEEE1588 with Prosilica GT cameras

## Description

IEEE1588 Precision Time Protocol (PTP) manages clock synchronization of multiple devices across an Ethernet network. Once the clocks of devices such as cameras, PCs, and sensors are synchronized, future software based triggers can be synchronized within 2 µs.

# Background

PTP was designed to improve on existing clock synchronization methods such as Network Time Protocol (NTP) and Global Positioning System (GPS). NTP suffers from poor accuracy, often quoted to be several milliseconds using a fast Ethernet network. GPS provides nanosecond precision using atomic clock and satellite triangulation; however it is an expensive component to incorporate into a camera.

PTP provides microsecond precision without increasing component cost, providing better accuracy than NTP at a lower cost than GPS.

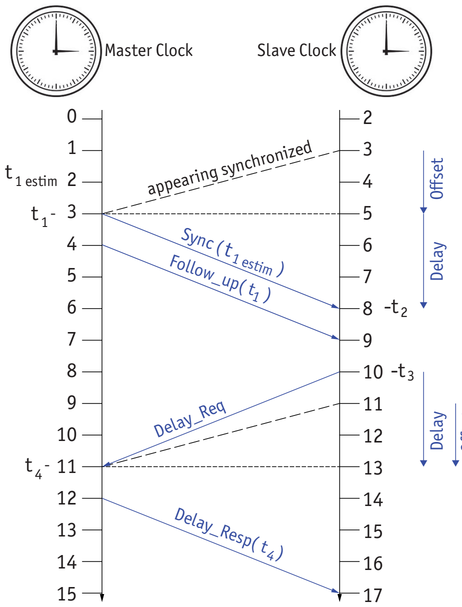


Figure 1: Multicast sync packets

### How is PTP synchronization achieved?

Synchronization begins when the device configured as the Master PTP clock transmits a “Sync” telegram using multicast messaging. Devices configured as Slave PTP clocks calculate the time difference between their clock and the Master PTP clock, and adjust accordingly.

Slave clock frequencies are constantly adjusted, through follow up and delay messages, to keep their clock value as close as possible to the master clock. While all Slave clocks are within 2 µs of the master, PTP sync is achieved.

### Network Topology

Achieving PTP sync between multiple devices requires that all devices be on the same network/subnet. This restriction is due to the current inability of any network card hardware to forward PTP sync multicast packets between ports within the 2 µs requirement.

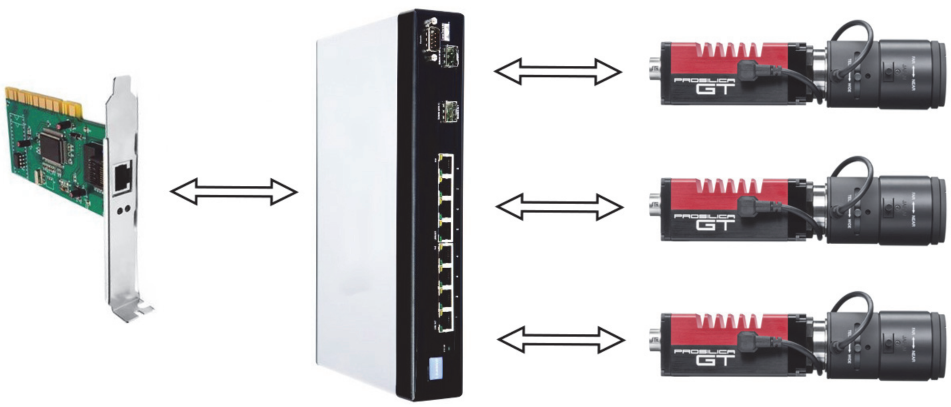

Figure 2: Multiple Allied Vision cameras on a switch

Therefore, all devices must be connected to a single port through a switch. Note that the total amount of bandwidth from all cameras must not exceed 125 MB on a single GigE connection, or packets will be dropped. Use **StreamBytesPerSecond** to reduce the bandwidth for each camera. Users of Allied Vision cameras requiring full GigE bandwidth from multiple cameras are encouraged to use a switch with a 10 GigE uplink and 10 GigE network card on the host.

**www**

For recommneded hardware see application note **Hardware_Selection_for_Allied Vision_GigE_Cameras.pdf**:
http://www.alliedvision.com/en/support/technical-papers-knowledge-base

# Enabling PTP synchronization with Prosilica GT cameras

PTP is enabled with the camera **PtpMode** attribute. Once a device's **PtpMode** is changed to *Master*, *Slave*, or *Auto*, synchronization of devices on the network begins to occur provided there is at minimum one master.

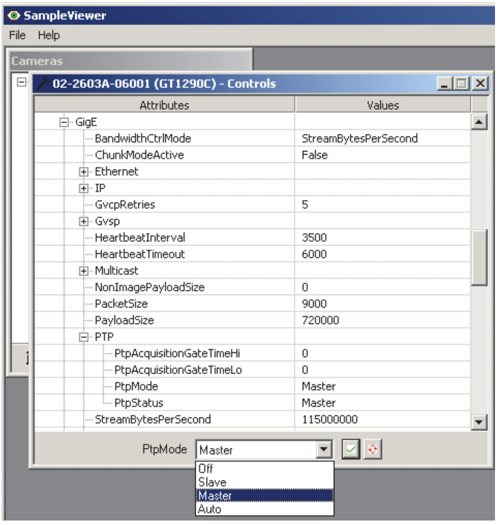


Figure 3: **PtpMode** attribute in SampleViewer

## PtpMode = Off

This camera's clock is not synchronized with any other device. This is the traditional functionality of a GigE camera. This is the factory preset PTP Mode on Allied Vision GigE cameras.

## PtpMode = Master

This camera's clock is the master clock. All other PTP enabled slave devices synchronize their clock to this camera.

## PtpMode = Slave

This camera's clock is altered to align with a Master device's clock.

## PtpMode = Auto

This camera uses the IEEE1588 best master clock algorithm to determine which device is master, and which devices are slaves. It may be assigned as either.

# Determining status of PTP synchronization

All Allied Vision cameras' device clocks are represented by the **TimeStampValue** attribute. **TimeStampValue** is divided into **TimeStampValueHi** and **TimeStampValueLo**. The period of each **TimeStampValueHi** count is ~ 4.295 seconds, the period of every **TimeStampValueLo** count is 1 ns.

**Note**

The actual resolution of Allied Vision cameras is 100 nanoseconds.

Synchronization is complete on all devices when all TimeStampValues are equal to the Master **TimeStampValue**, and no device's **PtpStatus** = *Syncing*, or **PtpStatus** = *Error*. **PtpStatus** is a read only attribute.

### PtpStatus = Off

The device **PtpMode** is set to off.

### PtpStatus = Master

PTP synchronization achieved. This device is acting as the master clock in the PTP network.

### PtpStatus = Slave

PTP synchronization achieved. This device is acting as a slave to another device's master clock.

### PtpStatus = Syncing

PTP synchronization not yet achieved but synchronization between master and slave(s) has started.

### PtpStatus = Error

PTP synchronization was previously achieved and has now failed.

## PTP Event Callbacks

In addition to **PtpStatus**, PvAPI users can use event callbacks to determine when synch is achieved, and if sync is ever lost. This is superior to polling **PtpStatus**.

**www**

PvAPI based source code demonstrating handling of PTP events callbacks:

https://alliedvisiontec.box.com/s/8937f0271b6e1a8fbcf7

```
Press 'q' to quit or 't' to set PtpAquisitionGateTime.
Cam[1]: 169.254.211.186 [GT1290C] opened
Cam[1]: PtpMode = Off.
Cam[1]: PtpStatus = Off
Cam[1]: PtpMode = Auto.
Cam[1]: Waiting for EventPtpSyncLocked
Cam[1]: PtpStatus = Syncing
Cam[2]: 169.254.22.193 [GT2750C] opened
Cam[2]: PtpMode = Off.
Cam[2]: PtpStatus = Off
Cam[2]: PtpMode = Auto.
Cam[2]: Waiting for EventPtpSyncLocked
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[2]: PtpStatus = Syncing
Cam[1]: PtpStatus = Syncing
Cam[1]: ***EventPtpSyncLocked
Cam[2]: PtpStatus = Syncing
Cam[2]: ***EventPtpSyncLocked
Cam[1]: ***EventAcquisitionStart
Cam[1]: AcquisitionStart success
Cam[2]: ***EventAcquisitionStart
Cam[2]: AcquisitionStart success
Cam[1]: Frame: [1] TimestampHi: [25] TimestampLo: [2162209400]
Cam[2]: Frame: [2] TimestampHi: [25] TimestampLo: [2162209500]
```

Figure 4: Output of running the **EventCallbackPTP** sample code

# PTP Synch Failure

In the event that a slave device clock drifts more than 2 µs from the master, **PtpStatus** is set to *Error* and a **EventPtpSyncLost** callback is sent (if setup).

The device must manually be reset to **PtpMode** = *Slave/Auto*. This is an intentional design choice – it is important for any application relying on Ptp synch to know if synch is lost.

The cause of a Ptp synch failure may be related to a network hardware component's inability to forward multicast sync packets successfully within the 2 µs required time.

# Executing a synchronized scheduled software trigger

Once all device clocks are synchronized, it is possible to issue a future simultaneous software trigger event to all Allied Vision cameras using the **PtpAcquisitionGateTime** attribute. Follow the steps below to enable a synchronized scheduled trigger on multiple GT cameras:

1. Achieve clock synchronization. Set **PtpMode** = *Master*, *Slave* or *Auto*.
2. Set camera **FrameRate**, **FrameStartTriggerMode**, and **AcquisitionMode** as desired (some possible scenarios listed below).
3. Call **AcquisitionStart**() to start an acquisition stream.
4. Determine the current camera **TimeStampValue** using **TimeStampValueLatch**.
5. Set **PtpAcquisitionGateTime** to a value which sufficiently exceeds the current camera **TimeStampValue**.

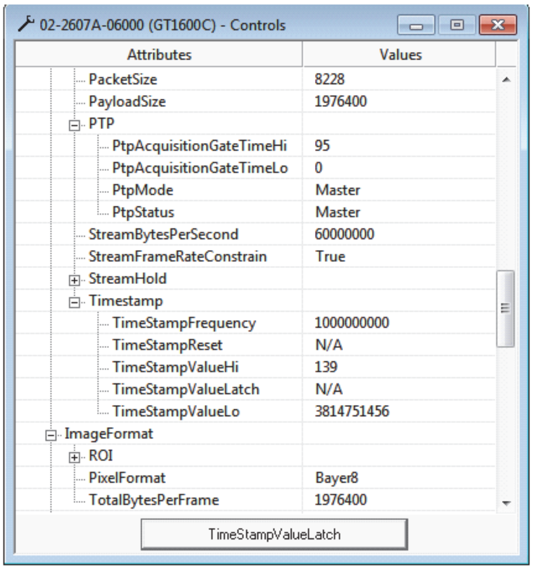


Figure 5: PTP attributes in the SampleViewer

$$\text{PtpAcquisitionGateTime} = \text{Current} \cdot \text{TimeStampValue} + \Delta$$

Where, Δ is the combined time required to set **PtpAcquisitionGateTime** on each camera (at minimum). This time will vary depending on OS, host, network hardware, and system load.

**Note**

The **EventCallbackPTP** example code:
https://alliedvisiontec.box.com/s/8937f0271b6e1a8fbcf7
This code calculates the time to set **PtpAcquisitionGateTime** for each camera. Running on an IntelR CoreTM2 Quad CPU, 2 GB RAM, Windows XP 32 bit, Intel Pro 1000 PT Dual network card, we measure 7 ms per camera.

Setting **PtpAcquisitionGateTime** will suspend all frame triggers in a camera's current **AcquisitionStart**() stream until **TimeStampValue** ≥ **PtpAcquisitionGateTime**, at which point frame triggers will resume.

# Typical use cases

## Case 1 - Fixed frame rate Synchronization

- Synchronize camera clocks
- Set **FrameStartTriggerMode** = *FixedRate* on all cameras
- Set **FrameRate** to the same value on all cameras
- Set **PtpAcquisitionGateTime** the same for all cameras and greater than existing **TimeStampValue**

When the camera **TimeStampValue** increments to the **PtpAcquisitionGateTime**, all cameras will start imaging synchronously at the configured frame rate. Reading the **TimeStampValue** contained in the image header will verify this synchronization.

## Case 2 - Precision timing by using GPS

Allied Vision cameras can be synchronized to a GPS timer, allowing "real world time" synchronization. Configure **PtpMode** on all of the cameras to *Slave* or *Auto*. In Auto, the IEEE1588 best master clock algorithm will elect the GPS clock as the master. Each camera will synchronize to the GPS master clock.

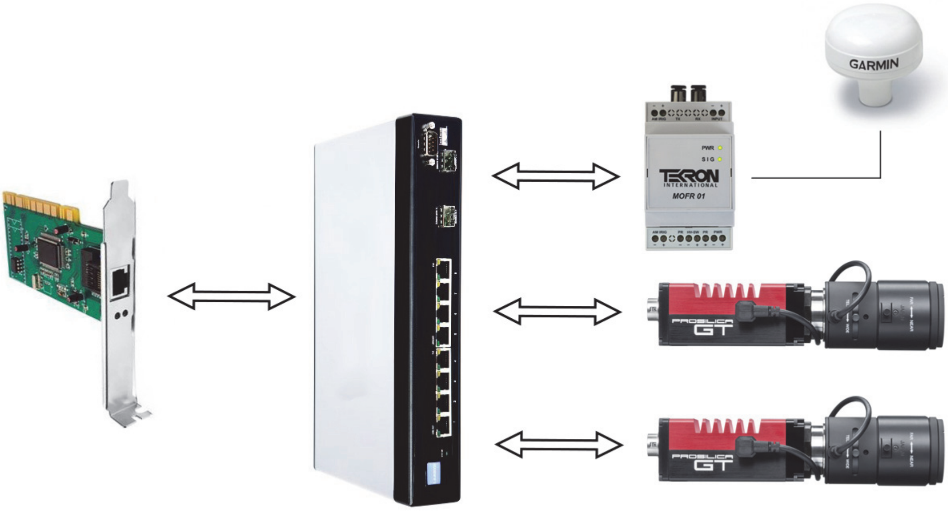


Figure 6: Synchronization using GPS clock

# Validating synchronization accuracy of PTP

## Method 1

PvApi users can view the **TimeStampValue** associated with any given frame through the **tPvFrame**->**Time-stampLo**

A total of 4 Allied Vision cameras were connected and set up in the following way.

- All cameras were configured to **FrameStartTriggerMode** = *FixedRate*, **FrameRate** = *1* fps
- **PtpMode** on one camera was set to Master, **PtpMode** on other three cameras set to *Slave*
- The same **PtpAcquisitionGateTime** was uploaded to all the cameras using approach shown in Case 1

The Graph below shows a histogram of the calculated offset between the recorded Master camera frame **TimeStampValue** and the **TimeStampValue** of the Slave cameras' **EventSyncInRise**. The data presented is compiled from 69 hours of continuous operation. The maximum offset recorded was less than ± 1 µs. This means that in this test, the timestamp of our three Slave cameras were synchronized to within 1 µs of the Master camera.

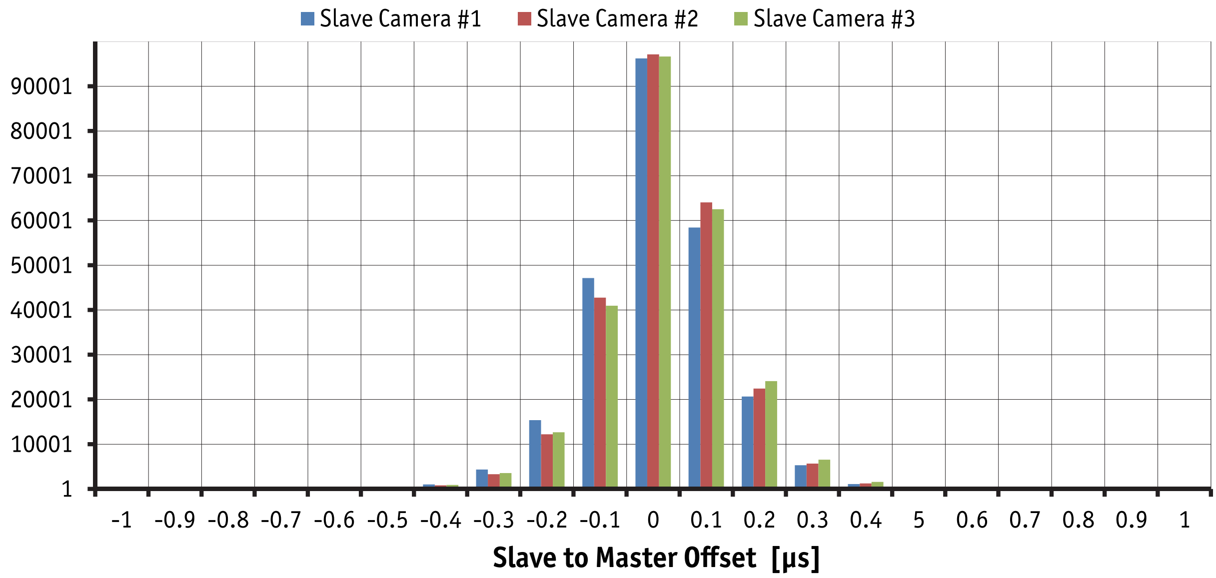


Figure 7: Histogram of Slave to Master Offset

## Method 2

To measure PTP accuracy in a different way we can use an oscilloscope and probe output triggers from multiple cameras. Two cameras were connected and setup in the following way:

- All cameras were configured to **FrameStartTriggerMode** = *FixedRate*, **FrameRate** = *15* fps
- **PtpMode** on one camera was set to Master, **PtpMode** on other camera set to *Slave*
- The same **PtpAcquisitionGateTime** was uploaded to all cameras using approach shown in Case 1
- The **SyncOut** signals on both cameras were set to *Strobe1*, **Strobe1ControlledDuration** = *On*, **Strobe1Duration** = *10 µs*, **Strobe1Mode** = *FrameTrigger*
- The SyncOut signals were monitored on a oscilloscope and viewed over a period of time

The capture below shows the readout from the oscilloscope when monitoring the SyncOut signals from both cameras. In the image below, channel 1 is approximately 125 ns offset from channel 2. This shows that the **TimeStampValue** synchronization between the two cameras is approximately 125 ns apart.

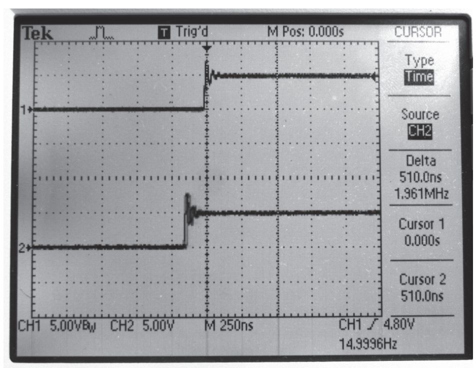


Figure 8: Scope Image of Oscilloscope showing trigger waveforms

# Frequently asked questions:

1. Can I use a host driven, software based PTP clock?

   Allied Vision has tested the possibility of using a PC based, PTP software clock to synchronize cameras across multiple adapters. In the process we discovered that using a host driven master clock on a non real time operating system such as windows or Linux generated very poor synchronization plagued by significant jitter between the timestamp of cameras. This approach may be sufficient for some applications but is does not generate the precise time synchronization we hope to achieve using PTP.

2. How do I translate **TimeStampValue** to real world time?

   Each **TimeStampValueHi** count is approximately 4.295 seconds, the period of every **TimeStampValueLo** count is 1 nanosecond, **TimeStampValue** frequency is 1 Ghz. You can use this to determine the value in seconds that the **TimeStampValue** represents. To associate this with real world time a GPS master clock needs to be used in the PTP network. Alternatively, you can reference the latest **TimeStampValue** to the PC clock however this will suffer from significant jitter.

# Additional References

GT Technical Manual, Allied Vision GigE camera and driver attributes:
http://www.alliedvision.com/en/support/technical-documentation

For technical support, please contact support@alliedvision.com.
For comments or suggestions regarding this document, please contact info@alliedvision.com.

# Disclaimer

Due to continual product development, technical specifications may be subject to change without notice. All trademarks are acknowledged as property of their respective owners. We are convinced that this information is correct. We acknowledge that it may not be comprehensive. Nevertheless, Allied Vision Technologies ("Allied Vision") cannot be held responsible for any damage in equipment or subsequent loss of data or whatsoever in consequence of this application note.