

Basler Components



Controlling Packet Transmission Timing with the Interpacket and Frame Transmission Delays on Basler GigE Vision Cameras

APPLICATION NOTES

Document Number: AW000649

Version: 02 Language: 000 (English)

Release Date: 30 July 2008



Contacting Basler Support Worldwide

Europe:

Basler AG
An der Strusbek 60 - 62
22926 Ahrensburg
Germany
Tel.: +49-4102-463-500
Fax.: +49-4102-463-599
bc.support.europe@baslerweb.com

Americas:

Basler, Inc.
855 Springdale Drive, Suite 160
Exton, PA 19341
U.S.A.
Tel.: +1-877-934-8472
Fax.: +1-610-280-7608
bc.support.usa@baslerweb.com

Asia:

Basler Asia Pte. Ltd
8 Boon Lay Way
03 - 03 Tradehub 21
Singapore 609964
Tel.: +65-6425-0472
Fax.: +65-6425-0473
bc.support.asia@baslerweb.com

www.baslerweb.com

**Copyright 2008 Basler Vision Technologies.
All material in this publication is subject to change without notice.**

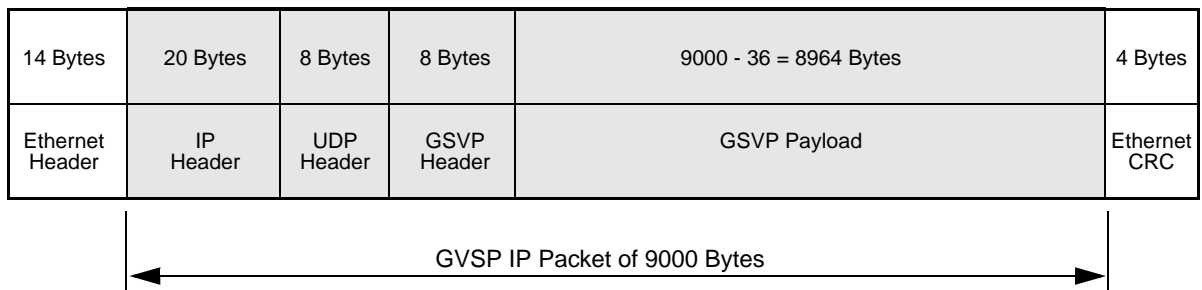
1 Introduction

Two parameters, frame start delay and interpacket delay, can be instrumental in controlling the orderly transmission of image data packets from basler cameras with a Gigabit Ethernet interface. These application notes describe some specific situations where the parameters are especially useful. They also point out a problem that you may encounter and provide a work around for the problem.

2 Details

The image data from Basler GigE Vision cameras (such as the scout, pilot, or runner) is transmitted in so-called GVSP IP packets from the camera to the PC. The size (in bytes) of those packets can be configured with the pylon Viewer or API. For a good quality network adapter (such as one from the Intel Pro 1000 family), a typical value would be 9000 bytes per packet.

The GigE Vision Streaming Protocol (GVSP) is based on the User Datagram Protocol (UDP). A GVSP IP packet with a packet size of 9000 bytes looks like this:



When data from a captured image is read out of the sensor, the camera does not wait until the whole image has been read out to start transmitting data. Instead, as soon as the data starts reading out of the sensor, the camera immediately starts building the data into IP packets of the configured size (e.g., 9000 bytes). As packets are completed, they are transmitted out of the camera over the GigE line. This process of creating and transmitting packets as the sensor is being read out guarantees minimal latency for image transmission.

Since the time it takes to read enough data out of the sensor to create a packet is typically longer than the time it takes to transmit the packet, there is a certain delay created between the transmission of any two packets.

To give an example:

According to the camera user's manual, the sensor readout time (for a whole frame) on a Basler scout scA780-54gm is 17962.4 μ s.

To determine the time needed for the readout of one packet, we must start by calculating the readout time for a single pixel. The sensor size for the scA780-54gm is 782 x 580 = 453560 pixels, so:

$$17962.4 \mu\text{s} \div 453560 \text{ pixels} = 0.03960314 \mu\text{s/pixel}$$

If we have configured the camera for an IP packet size of 9000 bytes, the size of one IP packet will actually be a bit bigger because we must add some Ethernet overhead:

$$9000 \text{ bytes} + 14 \text{ byte Ethernet header} + 4 \text{ byte Ethernet CRC} = 9018 \text{ bytes/packet}$$

However, the pixel data payload inside of one IP packet is only:

$$9000 \text{ bytes} - 20 \text{ byte IP header} - 8 \text{ byte UDP header} - 8 \text{ byte GVSP header} = 8964 \text{ bytes}$$

The readout time for the pixel data payload in an IP packet is determined by dividing the payload size by the bytes per pixel and then multiplying by the pixel readout time:

$$(8964 \text{ bytes} \div 1 \text{ byte/pixel}) \times 0.03960314 \mu\text{s/pixel} = 355.00254696 \mu\text{s}$$

Note that we are using 1 byte per pixel here. This would be true with the camera set to an 8 bit pixel format such as Mono 8.

Once enough pixel data has been read out of the sensor to create an IP packet, it takes some time to transmit the packet from the camera. On a Gigabit Ethernet network, you need 1 tick to transmit one byte (1 tick = 0.008 μ s = 8 ns). So the transmission time for one IP packet is:

$$9018 \text{ bytes} \times 0.008 \mu\text{s} = 72.144 \mu\text{s}$$

To determine the delay between the transmission of any two IP packets (i.e., the delay from the end of transmission of one packet to the start of transmission of the next), we must subtract the transmission time for an IP packet from the readout time for the pixel data payload:

$$355.00254696 \mu\text{s} - 72.144 \mu\text{s} = 282.85854696 \mu\text{s}$$

Or, in ticks:

$$282.85854696 \mu\text{s} \times (1 \text{ tick}/.008 \mu\text{s}) = 35357.31837 \text{ ticks}$$

The bandwidth that a camera will use to transmit acquired image data is typically controlled by setting the camera's packet size parameter and interpacket delay parameter. The interpacket delay parameter is used set the magnitude of the delay between the transmission of any two IP packets. These parameters can be set, for example, with the Basler pylon viewer or API.

But even if a Basler GigE Vision camera has been configured for an interpacket delay of zero, you will still see a "native" interpacket delay between the transmission of any two IP packets caused by the fact that reading image data out of the sensor takes longer than transmitting the packetized data! On the scA780-54gm used in our example above, the native delay will be about 35357 ticks.

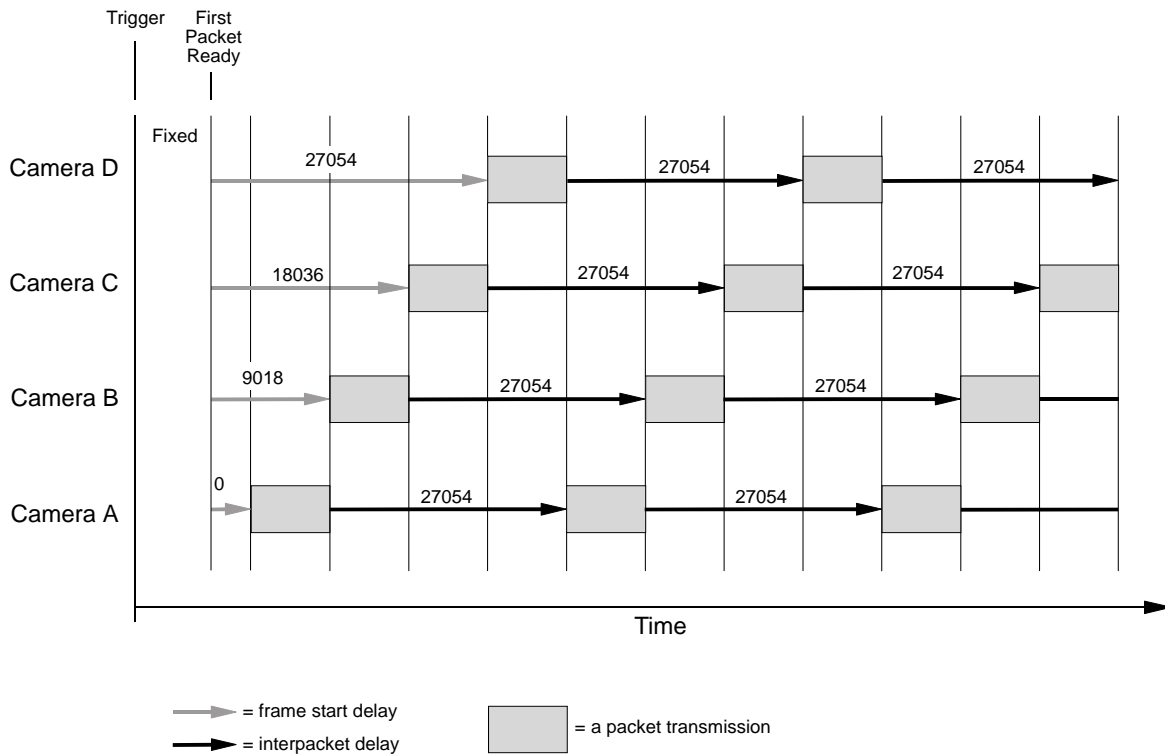
This means that if the camera's interpacket delay parameter setting is less than the native interpacket delay, the interpacket delay parameter will have no effect on the bandwidth used by the camera. **When the interpacket delay is set lower than the native delay, you have no real control over the delay between packet transmissions.** (Keep in mind that the bandwidth a camera is using can also have an impact on the camera's frame rate.)

In some situations, however, better control over the timing of packet transmission is required. This is particularly true, if you have multiple cameras connected to one switch, and you trigger all of the cameras simultaneously. To avoid a situation where all of the cameras transmit packets at the same time and all of the packets arrive at your network switch at the same time, you might consider configuring your camera with a frame transmission delay and an interpacket delay.

First, let's look at the definition of the camera's frame transmission and interpacket delay parameters:

- When a frame capture is triggered, there is a "fixed" delay between the time when the frame is triggered and when the camera is ready to transmit the first IP packet for the frame. This fixed delay includes the time it takes to expose the sensor, plus the time it takes to begin reading out the sensor, plus the time it takes to prepare the first packet. Normally, the camera would start to transmit the first packet as soon as the packet is ready. The **Frame Start Delay** parameter sets a delay time (in ticks) between when the camera is ready to start transmitting the first packet and when it actually will start transmitting the packet. For example, if the delay is set to zero, the camera will transmit the first packet as soon as it is ready. If the delay is set to 1000, when the camera is ready to transmit the first packet, it will wait for 1000 ticks before it actually starts to transmit the packet. Note that the frame transmission delay only applies to the first packet for the frame.
- The **Interpacket Delay** parameter sets a delay time between the end of transmission of each packet for the frame and the beginning of transmission of the next packet for the frame. For example, if the camera has just finished transmitting the first packet for a frame and the interpacket delay is set to 20000 ticks, the camera will delay 20000 ticks before it begins transmitting the second packet.

So if you wanted to trigger four cameras (A,B,C, and D) simultaneously and if you wanted to avoid having the data packets transmitted from the camera arriving at your network switch at the same time, you could consider configuring the cameras with a different frame transmission delay for each camera and a single interpacket delay as illustrated in the figure on the next page.



The figure above assumes that you are using four Basler scout scA780-54gm cameras (designated as camera A, B, C and D), that the cameras are set for the Mono 8 pixel format, and that the packet size parameter on each camera has been set to 9018 bytes. Knowing that in Mono 8 each pixel is one byte and that it takes one tick to transmit a byte, you might think that it is a good idea to set the cameras this way:

- frame transmission delay for camera A = 0
- frame transmission delay for camera B = 9018 ticks
- frame transmission delay for camera C = 18036 ticks (i.e., 9018×2 ticks)
- frame transmission delay for camera D = 27054 ticks (i.e., 9018×3 ticks)
- to allow cameras B, C, and D to send their packets during the interpacket delay period for camera A, you would need an interpacket delay of 9018×3 or 27054 ticks
- all four cameras would be set with this interpacket delay

While this analysis of the problem makes perfect sense, **it will not work!** The problem is that an interpacket delay setting of 27054 ticks is below the 35357 tick "native" interpacket delay that we calculated earlier for scA780-54gm cameras.

To work around this situation, you should set the frame transmission delay for camera A to ensure that the transmission of the first packet for the frame will end at just the moment when enough data has been read out of the sensor to create packet two. To find this point, we recommend that you set the interpacket delay for camera A to the theoretical optimum (in our example, this would be 27054 ticks) and then you gradually increase the frame start delay until you start to see a decrease in the camera's frame rate.

By using this procedure, we found that a good frame transmission delay for camera A in our example would be 60000 ticks. So a setup that would work well would look like this:

- frame transmission delay for camera A = 60000 ticks
- frame transmission delay for camera B = 69018 ticks (i.e., 60000 + 9018)
- frame transmission delay for camera C = 78036 ticks (i.e., 60000 + 18036)
- frame transmission delay for camera B = 87054 ticks (i.e., 60000 + 27054)

With these frame transmission delays, you could use the optimum 27054 tick interpacket delay. All four cameras would be set with this interpacket delay.

Revision History

Doc. ID Number	Date	Changes
AW00064901000	8 Apr 2008	Initial release of this document.
AW00064901000	30 Jul 2008	Updated contact addresses and phone numbers.