APPLICATION NOTE

Basler GigE Vision Network Drivers and Bandwidth Management

Document Number: AW001445
Version: 01 Language: 000 (English)
Release Date: 10 July 2017

BASLER
the power of sight

Contacting Basler Support Worldwide

Europe, Middle East, Africa

Basler AG
An der Strusbek 60–62
22926 Ahrensburg
Germany

Tel. +49 4102 463 515
Fax +49 4102 463 599

support.europe@baslerweb.com

The Americas

Basler, Inc.
855 Springdale Drive, Suite 203
Exton, PA 19341
USA

Tel. +1 610 280 0171
Fax +1 610 280 7608

support.usa@baslerweb.com

Asia-Pacific

Basler Asia Pte. Ltd.
35 Marsiling Industrial Estate Road 3
#05–06
Singapore 739257

Tel. +65 6367 1355
Fax +65 6367 1255

support.asia@baslerweb.com

www.baslerweb.com

Table of Contents

1 About this Document

The content of this document is taken from the legacy *ace GigE User's Manual* (AW000893), appendix A.1, A.2, and B.2.

For more information about ace GigE cameras and network related parameters, see the *Basler Product Documentation* (AW001406).

2 Basler GigE Vision Network Drivers and Parameters

This section describes the network drivers available for your Basler GigE camera and provides detailed information about the parameters associated with the drivers.

Three network drivers are available for the network adapter used with your GigE cameras:

- The **Basler Filter Driver** is a basic GigE Vision network driver that is compatible with all network adapters. The advantage of this driver is its extensive compatibility.
- The **Basler Performance Driver** is a hardware-specific GigE Vision network driver. The driver is only compatible with network adapters that use specific Intel chipsets. The advantage of the performance driver is that it significantly lowers the CPU load needed to service the network traffic between the computer and the camera(s). It also has a more robust packet resend mechanism.
- The **Basler Socket Driver** is not a real driver, but uses the Socket API of the given operating system, e.g., Windows, Linux, or Mac OS X, to communicate with cameras instead. The advantage of the socket driver is that it does not need any installation and is compatible with all network adapters. When using the socket driver, Basler recommends to adjust the network adapter settings (e.g., optimize the use of Jumbo Frames, Receive Descriptors, and Interrupt Moderation Rate) as described in the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611), section 4.3.1.

For more information about compatible Intel chipsets and installing the network drivers, see the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611)

2.1 Basler Filter Driver

The Basler Filter Driver is a basic driver GigE Vision network driver. It is designed to be compatible with most network adapter cards.

The functionality of the filter driver is relatively simple. For each frame, the driver checks the order of the incoming packets. If the driver detects that a packet or a group of packets is missing, it waits for a specified period of time to see if the missing packet or group of packets arrives. If the packet or group does not arrive within the specified period, the driver sends a resend request for the missing packet or group of packets.

The parameters associated with the filter driver are described below.

EnableResend - Enables or disables the packet resend mechanism.

If packet resend is disabled and the filter driver detects that a packet has been lost during transmission, the grab result for the returned buffer holding the image indicates that the grab failed and the image is incomplete.

If packet resend is enabled and the driver detects that a packet has been lost during transmission, the driver sends a resend request to the camera. If the camera still has the packet in its buffer, it resends the packet. If there are several lost packets in a row, the resend requests are combined.

PacketTimeout - The PacketTimeout parameter defines how long (in milliseconds) the filter driver waits for the next expected packet before it initiates a resend request. Make sure the PacketTimeout parameter is set to a longer time interval than the time interval set for the interpacket delay.

FrameRetention - The FrameRetention parameter sets the timeout (in milliseconds) for the frame retention timer. Whenever the filter driver detects the leader for a frame, the frame retention timer starts. The timer resets after each packet in the frame is received and times out after the last packet is received. If the timer times out at any time before the last packet is received, the buffer for the frame is released and indicated as an unsuccessful grab.

You can set the filer driver parameter values from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to read and write the parameter values:

```
// Enable Resend
camera_t::StreamGrabber_t StreamGrabber (Camera.GetStreamGrabber(0));
StreamGrabber.EnableResend.SetValue(false); // disable resends

// Packet Timeout/FrameRetention
camera_t::StreamGrabber_t StreamGrabber (Camera.GetStreamGrabber(0));
StreamGrabber.PacketTimeout.SetValue(40);
StreamGrabber.FrameRetention.SetValue(200);
```

You can also use the Basler pylon Viewer application to easily set the parameters.

2.2 Basler Performance Driver

The Basler Performance Driver is a hardware-specific GigE Vision network driver compatible with the following network adapters:

- Intel Pro 1000 series
- Intel i210 series (formerly "Springville")
- Intel i340 series
- Intel i350 series

These adapters generally work well with the Basler performance driver. However, since the Intel Pro 1000 series has changed over the time, it may happen that the Basler performance driver does not support your particular Intel Pro 1000 adapter. Contact Basler technical support for recommendations of currently available Pro 1000 adapters and for information about compatible chipsets.

The main advantage of the performance driver is that it significantly lowers the CPU load needed to service the network traffic between the computer and the camera(s). It also has a more robust packet resend mechanism.

The performance driver uses two distinct "resend mechanisms" to trigger resend requests for missing packets:

- Threshold resend mechanism
- Timeout resend mechanism

The mechanisms are independent from each other and can be used separately. However, for maximum efficiency and to ensure that resend requests are sent for all missing packets, Basler recommends using both resend mechanisms in a specific, optimized combination, as provided by the parameter default values.

The performance driver's parameter values determine how the resend mechanisms act and how they relate to each other. You can set the parameter values by using the pylon Viewer or from within your application software by using the pylon API.

The parameter default values provide for the following:

- The threshold resend mechanism precedes the timeout resend mechanism. This ensures that a resend request is sent for every missing packet, even at very high rates of arriving packets.
- The timeout resend mechanism is effective for those missing packets that were not resent after the first resend request.

Basler strongly recommends using the default parameter settings. Only users with the necessary expertise should change the default parameter values.

The Basler Performance Driver uses a "receive window" to check the status of packets. The driver checks for missing packets as packets enter the receive window. If a packet arrives from higher in the sequence of packets than expected, the preceding skipped packet or packets are detected as missing. For example, suppose packet (n-1) has entered the receive window and is immediately

followed by packet (n+1). In this case, as soon as packet (n+1) enters the receive window, packet n is detected as missing.

2.2.1 General Parameters

EnableResend - Enables the packet resend mechanisms.

If the EnableResend parameter is set to false, the resend mechanisms are disabled. The performance driver does not check for missing packets and does not send resend requests to the camera.

If the Enable Resend parameter is set to true, the resend mechanisms are enabled. The performance driver checks for missing packets. Depending on the parameter settings and the resend response, the driver sends one or several resend requests to the camera.

ReceiveWindowSize - Sets the size of the receive window.

2.2.2 Threshold Resend Mechanism Parameters

The threshold resend request mechanism is illustrated in Figure 1 where the following assumptions are made:

- Packets 997, 998, and 999 are missing from the stream of packets.
- Packet 1002 is missing from the stream of packets.

DIAGRAM IS NOT DRAWN TO SCALE

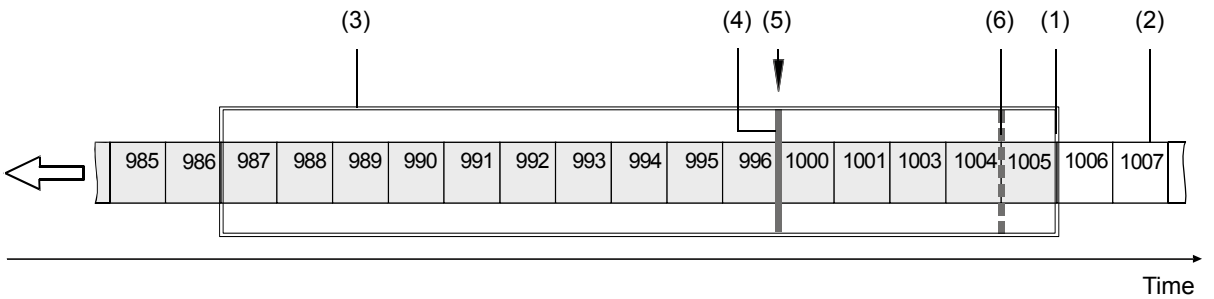

Fig. 1: Example of a Receive Window with Resend Request Threshold & Resend Request Batching Threshold

(1) Front end of the receive window. Missing packets are detected here.

(2) Stream of packets. Gray indicates that the status was checked as the packet entered the receive window. White indicates that the status has not yet been checked.

(3) Receive window of the performance driver.

(4) Threshold for sending resend requests (resend request threshold).

(5) A separate resend request is sent for each packets 997, 998, and 999.

(6) Threshold for batching resend requests for consecutive missing packets (resend request batching threshold). Only one resend request is sent for the consecutive missing packets.

ResendRequestThreshold - This parameter determines the location of the resend request threshold within the receive window as shown in Figure 1. The parameter value is in per cent of the width of the receive window. In Figure 1 the resend request threshold is set at 33.33% of the width of the receive window.

A stream of packets advances packet by packet beyond the resend request threshold (i.e. to the left of the resend request threshold in Figure 1). As soon as the position where a packet is missing advances beyond the resend request threshold, a resend request is sent for the missing packet.

In the example shown in Figure 1, packets 987 to 1005 are within the receive window and packets 997 to 999 and 1002 were detected as missing. In the situation shown, a resend request is sent to the camera for each of the missing consecutive packets 997 to 999. The resend requests are sent after packet 996 - the last packet of the intact sequence of packets - has advanced beyond the resend request threshold and before packet 1000 - the next packet in the stream of packets - can advance beyond the resend request threshold. Similarly, a resend request is sent for missing packet 1002 after packet 1001 has advanced beyond the resend request threshold and before packet 1003 can advance beyond the resend request threshold.

ResendRequestBatching - This parameter determines the location of the resend request batching threshold in the receive window (Figure 1). The parameter value is in per cent of a span that starts with the resend request threshold and ends with the front end of the receive window. The maximum allowed parameter value is 100. In Figure 1 the resend request batching threshold is set at 80% of the span.

The resend request batching threshold relates to consecutive missing packets, i.e., to a continuous sequence of missing packets. Resend request batching allows grouping of consecutive missing packets for a single resend request rather than sending a sequence of resend requests where each resend request relates to just one missing packet.

The location of the resend request batching threshold determines the maximum number of consecutive missing packets that can be grouped together for a single resend request. The maximum number corresponds to the number of packets that fit into the span between the resend request threshold and the resend request batching threshold plus one.

If the ResendRequestBatching parameter is set to 0, no batching occurs and a resend request is sent for each single missing packet. For other settings, consider an example: Suppose the Resend Request Batching parameter is set to 80 referring to a span between the resend request threshold and the front end of the receive window that can hold five packets (Figure 1). In this case, 4 packets (5 x 80%) fit into the span between the resend request threshold and the resend request batching threshold. Accordingly, the maximum number of consecutive missing packets that can be batched is 5 (4 + 1).

2.2.3 Timeout Resend Mechanism Parameters

The timeout resend mechanism is illustrated in Figure 2 where the following assumptions are made:

- The frame includes 3000 packets.
- Packet 1002 is missing within the stream of packets and has not been recovered.
- Packets 2999 and 3000 are missing at the end of the stream of packets (end of the frame).
- The Maximum Number Resend Requests parameter is set to 3.

DIAGRAM IS NOT DRAWN TO SCALE

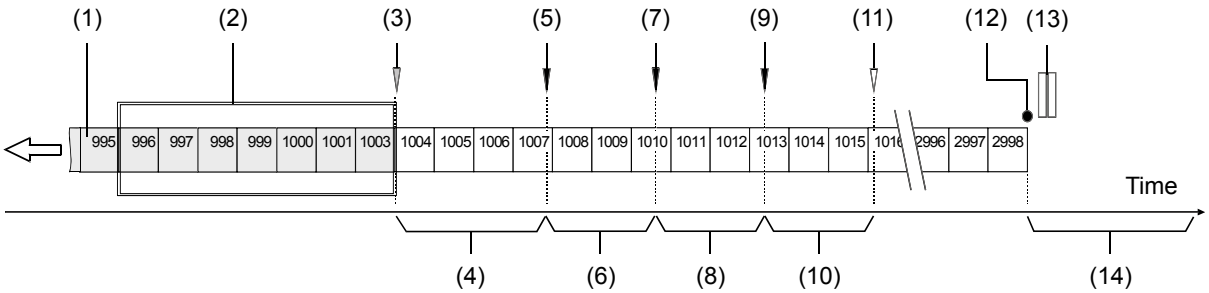

Fig. 2: Incomplete Stream of Packets and Part of the Resend Mechanism

(1) Stream of packets. Gray indicates that the status was checked as the packet entered the receive window. White indicates that the status has not yet been checked.

(2) Receive window of the performance driver.

(3) As packet 1003 enters the receive window, packet 1002 is detected as missing.

(4) Interval defined by the ResendTimeout parameter.

(5) The resend timeout interval expires and the first resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(6) Interval defined by the ResendResponseTimeout parameter.

(7) The resend response timeout interval expires and a second resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(8) Interval defined by the ResendResponseTimeout parameter.

(9) The resend response timeout interval expires and a third resend request for packet 1002 is sent to the camera. The camera still does not respond with a resend.

(10) Interval defined by the ResendResponseTimeout parameter.

(11) Because the maximum number of resend requests has been sent and the last resend responset timeout interval has expired, packet 1002 is now considered as lost.

(12) End of the frame.

(13) Missing packets at the end of the frame (2999 and 3000).

(14) Interval defined by the PacketTimeout parameter.

MaximumNumberResendRequests - The MaximumNumberResendRequests parameter sets the maximum number of resend requests the performance driver sends to the camera for each missing packet.

ResendTimeout - The ResendTimeout parameter defines how long (in milliseconds) the performance driver waits after detecting that a packet is missing before sending a resend request to the camera. The parameter applies only once to each missing packet after the packet was detected as missing.

ResendRequestResponseTimeout - The ResendRequestResponseTimeout parameter defines how long (in milliseconds) the performance driver waits after sending a resend request to the camera before considering the resend request as lost.

If a resend request for a missing packet is considered lost and if the maximum number of resend requests as set by the MaximumNumberResendRequests parameter has not yet been reached, another resend request is sent. In this case, the parameter defines the time separation between consecutive resend requests for a missing packet.

PacketTimeout - The PacketTimeout parameter defines how long (in milliseconds) the performance driver waits for the next expected packet before it sends a resend request to the camera. This parameter ensures that resend requests are sent for missing packets near to the end of a frame. In the event of a major interruption in the stream of packets, the parameter also ensures that resend requests are sent for missing packets that were detected to be missing immediately before the interruption. Make sure the Packet Timeout parameter is set to a longer time interval than the time interval set for the inter-packet delay.

2.2.4 Threshold and Timeout Resend Mechanisms Combined

Figure 3 illustrates the combined action of the threshold and the timeout resend mechanisms where the following assumptions are made:

- All parameters set to default.
- The frame includes 3000 packets.
- Packet 1002 is missing within the stream of packets and has not been recovered.
- Packets 2999 and 3000 are missing at the end of the stream of packets (end of the frame).

The default values for the performance driver parameters let the threshold resend mechanism become operative before the timeout resend mechanism. This ensures maximum efficiency and that resend requests are sent for all missing packets.

With the default parameter values, the resend request threshold is located very close to the front end of the receive window. Accordingly, there is only a minimum delay between detecting a missing packet and sending a resend request for it. In this case, a delay according to the Resend Timeout parameter does not occur (see Figure 3). In addition, resend request batching does not occur.

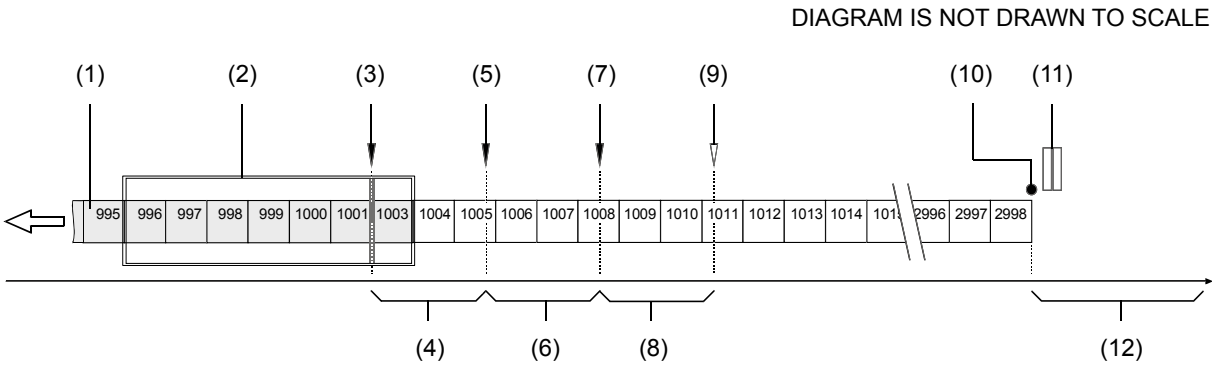

Fig. 3: Combination of Threshold Resend Mechanism and Timeout Resend Mechanism

(1) Stream of packets, Gray indicates that the status was checked as the packet entered the receive window. White indicates that the status has not yet been checked.

(2) Receive window of the performance driver.

(3) Threshold for sending resend requests (resend request threshold). The first resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(4) Interval defined by the ResendResponseTimeout parameter.

(5) The ResendTimeout interval expires and the second resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(6) Interval defined by the ResendResponseTimeout parameter

(7) The resend timeout interval expires and the third resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(8) Interval defined by the ResendResponseTimeout parameter

(9) Because the maximum number of resend requests has been sent and the last resend response timeout interval has expired, packet 1002 is now considered as lost.

(10) End of the frame.

(11) Missing packets at the end of the frame (2999 and 3000).

(12) Interval defined by the PacketTimeout parameter.

You can set the performance driver parameter values from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to read and write the parameter values:

```
// Get the Stream Parameters object
Camera_t::StreamGrabber_t StreamGrabber(Camera.GetStreamGrabber(0));

// Write the ReceiveWindowSize parameter
StreamGrabber.ReceiveWindowSize.SetValue(16);

// Disable packet resends
StreamGrabber.EnableResend.SetValue(false);
```

```
// Write the PacketTimeout parameter
StreamGrabber.PacketTimeout.SetValue(40);

// Write the ResendRequestThreshold parameter
StreamGrabber.ResendRequestThreshold.SetValue(5);

// Write the ResendRequestBatching parameter
StreamGrabber.ResendRequestBatching.SetValue(10);

// Write the ResendTimeout parameter
StreamGrabber.ResendTimeout.SetValue(2);

// Write the ResendRequestResponseTimeout parameter
StreamGrabber.ResendRequestResponseTimeout.SetValue(2);

// Write the MaximumNumberResendRequests parameter
StreamGrabber.MaximumNumberResendRequests.SetValue(25);
```

You can also use the Basler pylon Viewer application to easily set the parameters.

The performance driver parameters only appear in the pylon Viewer if the performance driver is installed on the adapter to which your camera is connected.

2.2.5 Adapter Properties

When the Basler Performance driver is installed, it adds a set of "advanced" properties to the network adapter. These properties include:

Max Packet Latency - A value in microseconds that defines how long the adapter waits after it receives a packet before it generates a packet received interrupt.

Max Receive Inter-Packet Delay - A value in microseconds that defines the maximum amount of time allowed between incoming packets.

Maximum Interrupts per Second - Sets the maximum number of interrupts per second that the adapter generates.

Network Address - Allows to specify a MAC address that overrides the default address provided by the adapter.

Packet Buffer Size - Sets the size in bytes of the buffers used by the receive descriptors and the transmit descriptors.

Receive Descriptors - Sets the number of descriptors to use in the adapter’s receiving ring.

Transmit Descriptors - Sets the number of descriptors to use in the adapter’s transmit ring.

To access the advanced properties for an adapter:

1. Open a **Network Connections** window and find the connection for your network adapter.
2. Right click on the name of the connection and select **Properties** from the drop down menu.
   A **LAN Connection Properties** window opens.
3. Click the **Configure** button.
   An **Adapter Properties** window opens.
4. Click the **Advanced** tab.

	**We strongly recommend using the default parameter settings.** Changing the parameters can have a significant negative effect on the performance of the adapter and the driver.

2.2.6 Transport Layer Parameters

The transport layer parameters are part of the camera's basic GigE implementation. These parameters do not normally require adjustment.

ReadTimeout - If a register read request is sent to the camera via the transport layer, this parameter designates the time out (in milliseconds) within which a response must be received.

WriteTimeout - If a register write request is sent to the camera via the transport layer, this parameter designates the time out (in milliseconds) within which an acknowledge must be received.

HeartbeatTimeout - The GigE Vision standard requires implementation of a heartbeat routine to monitor the connection between the camera and the host computer. This parameter sets the heartbeat timeout (in milliseconds). If a timeout occurs, the camera releases the network connection and enters a state that allows reconnection.

	Management of the heartbeat time is normally handled by the Basler's basic GigE implementation and changing this parameter is not required for normal camera operation. However, if you are debugging an application and you stop at a break point, you will have a problem with the heartbeat timer. The timer will time out when you stop at a break point and the connection to the camera will be lost. When debugging, you should increase the heartbeat timeout to a high value to avoid heartbeat timeouts at break points. When debugging is complete, you should return the timeout to its normal setting.

You can set the driver related transport layer parameter values from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to read and write the parameter values:

```
// Read/Write Timeout
Camera_t::TlParams_t TlParams( Camera.GetTLNodeMap() );
TlParams.ReadTimeout.SetValue(500);  // 500 milliseconds
TlParams.WriteTimeout.SetValue(500); // 500 milliseconds

// Heartbeat Timeout
Camera_t::TlParams_t TlParams( Camera.GetTLNodeMap() );
TlParams.HeartbeatTimeout.SetValue(5000);  // 5 seconds
```

You can also use the Basler pylon Viewer application to easily set the parameters.

3 Managing Bandwidth With Multiple Cameras

3.1 Overview

If you are using a single camera on a GigE network, the problem of managing bandwidth is simple. The network can easily handle the bandwidth needs of a single camera and no intervention is required. A more complicated situation arises if you have multiple cameras connected to a single network adapter as shown in Figure 4.

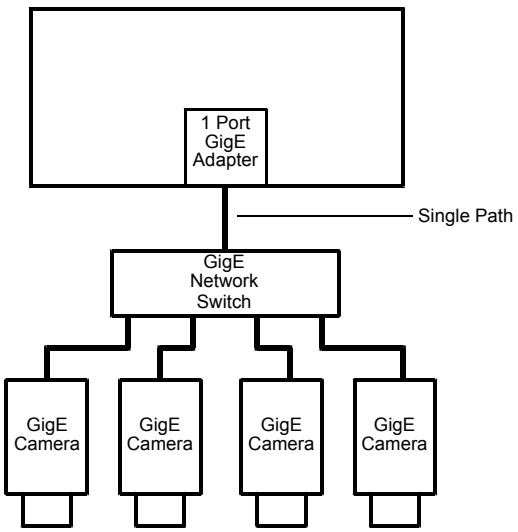

Fig. 4: Multiple Cameras on a Network

One way to manage the situation where multiple cameras are sharing a single network path is to make sure that only one of the cameras is acquiring and transmitting images at any given time. The data output from a single camera is well within the bandwidth capacity of the single path and you should have no problem with bandwidth in this case.

If you want to acquire and transmit images from several cameras simultaneously, however, you must determine the total data output rate for all the cameras that will be operating simultaneously and you must make sure that this total does not exceed the bandwidth of the single path (125 MByte/s).

An easy way to make a quick check of the total data output from the cameras that will operate simultaneously is to read the value of the BandwidthAssigned parameter for each camera. This

parameter indicates the camera's gross data output rate in bytes per second with its current settings. If the sum of the bandwidth assigned values is less than 125 MByte/s, the cameras should be able to operate simultaneously without problems. If it is greater, you must lower the data output rate of one or more of the cameras.

You can lower the data output rate on a camera by using the Inter-PacketDelay parameter. This parameter adds a delay between the transmission of each packet from the camera and thus slows the data transmission rate of the camera. The higher the inter-packet delay parameter is set, the greater the delay between the transmission of each packet will be and the lower the data transmission rate will be. After you have adjusted the Inter-PacketDelay parameter on each camera, you can check the sum of the BandwidthAssigned parameter values and see if the sum is now less than 125 MByte/s.

3.2 A Procedure for Managing Bandwidth

In theory, managing bandwidth sharing among several cameras is as easy as adjusting the inter-packet delay. In practice, it is a bit more complicated because you must consider several factors when managing bandwidth. The procedure below outlines a structured approach to managing bandwidth for several cameras.

The objectives of the procedure are:

- To optimize network performance.
- To determine the bandwidth needed by each camera for image data transmission.
- To determine the bandwidth actually assigned to each camera for image data transmission.
- For each camera, to make sure that the actual bandwidth assigned for image data transmission matches the bandwidth needed.
- To make sure that the total bandwidth assigned to all cameras does not exceed the network's bandwidth capacity.
- To make adjustments if the bandwidth capacity is exceeded.

3.2.1 Step 1: Improving the Network Performance

If you use, as recommended, the Basler Performance Driver with a compatible network adapter (see Section 2.2 on page 4), the network parameters for the network adapter are automatically optimized and need not be changed.

If you use the filter driver and have already set network parameters for your network adapter during the installation of the Basler pylon software, continue with step two. Otherwise, open the **Network Connection Properties** window for your network adapter and check the following network parameters:

- If you use a compatible network adapter: Make sure the **ReceiveDescriptors** parameter is set to its maximum value and the **InterruptModerationRate** parameter is set to **Extreme**.

Also make sure the **Speed and Duplex Mode** parameter is set to **Auto Detect**.

- If you use a different network adapter, see whether parameters are available that allow setting the number of receive descriptors and the number of CPU interrupts. The related parameter names may differ from the ones used for the compatible network adapters. Also, the way of setting the parameters may be different. You may, e.g., have to use a parameter to set a low number for the interrupt moderation and then use a different parameter to enable the interrupt moderation.

  If possible, set the number of receive descriptors to a maximum value and set the number of CPU interrupts to a low value.

  If possible, also set the parameter for speed and duplex to auto.

Contact Basler technical support if you need further assistance.

3.2.2 Step 2: Setting the PacketSize Parameter As Large as Possible

Using the largest possible packet size has two advantages, it increases the efficiency of network transmissions between the camera and the computer and it reduces the time required by the computer to process incoming packets. The largest packet size setting that you can use with your camera is determined by the largest packet size that can be handled by your network. The size of the packets that can be handled by the network depends on the capabilities and settings of the network adapter you are using and on capabilities of the network switch you are using.

Unless you have already set the packet size for your network adapter during the installation of the Basler pylon software, check the documentation for your adapter to determine the maximum packet size (sometimes called “frame” size) that the adapter can handle. Many adapters can handle what is known as “jumbo packets” or "jumbo frames". These are packets with a maximum size of 9 kB. Once you have determined the maximum size packets the adapter can handle, make sure that the adapter is set to use the maximum packet size.

Next, check the documentation for your network switch and determine the maximum packet size that it can handle. If there are any settings available for the switch, make sure that the switch is set for the largest packet size possible.

Now that you have set the adapter and switch, you can determine the largest packet size the network can handle. The device with the smallest maximum packet size determines the maximum allowed packet size for the network. For example, if the adapter can handle 8 kB packets and the switch can handle 6 kB packets, then the maximum for the network is 6 kB packets.

Once you have determined the maximum packet size for your network, set the value of the Packet Size parameter on each camera to this value.

The manufacturer's documentation sometimes makes it difficult to determine the maximum packet size for a device, especially network switches. There is a "quick and dirty" way to check the maximum packet size for your network with its current configuration:

1. Open the pylon Viewer, select a camera, and set the PacketSize parameter to a low value (1 kB for example).
2. Use the continuous shot mode to capture several images.
3. Gradually increase the value of the PacketSize parameter and capture a few images after each size change.
4. When your packet size setting exceeds the packet size that the network can handle, the viewer loses the ability to capture images. (When you use continuous shot, the viewer's status bar indicates that it is acquiring images, but the image in the viewing area appears to be frozen.)

3.2.3 Step 3: Setting the BandwidthReserve Parameter

The BandwidthReserve parameter setting for a camera determines how much of the bandwidth assigned to that camera is reserved for lost packet resends and for asynchronous traffic such as commands sent to the camera. If you are operating the camera in a relatively EMI free environment, you may find that a bandwidth reserve of 2% or 3% is adequate. If you are operating in an extremely noisy environment, you may find that a reserve of 8 % or 10 % is more appropriate.

3.2.4 Step 4: Calculating the Data Bandwidth Needed

The objective of this step is to determine how much bandwidth (in Byte/s) each camera needs to transmit the image data that it generates. The amount of data bandwidth a camera needs is the product of several factors: the amount of data included in each image, the amount of chunk data being added to each image, the "packet overhead" such as packet leaders and trailers, and the number of frames the camera is acquiring each second.

For each camera, you can use the two formulas below to calculate the data bandwidth needed. To use the formulas, you need to know the current value of the PayloadSize parameter and the PacketSize parameter for each camera. You also need to know the frame rate (in frames/s) at which each camera will operate.

$$\text{Bytes/Frame} = \left[\left\lceil \frac{\text{Payload Size}}{\text{Packet Size}} \right\rceil^{1} \times \text{Packet Overhead} \right] + \lceil \text{Payload Size} \rceil^{4} + \text{Leader Size} + \text{Trailer Size}$$

Data Bandwidth Needed = Bytes/Frame x Frames/s

Where:

Packet Overhead =72 (for a GigE network)

78 (for a 100 MBit/s network)

Leader Size =Packet Overhead + 36 (if chunk mode is not active)

Packet Overhead + 12 (if chunk mode is active)

Trailer Size = Packet Overhead + 8

$\lceil x \rceil_1$ means round up x to the nearest integer

$\lceil x \rceil_4$ means round up x to the nearest multiple of 4

Step 5 - Calculate “data bandwidth assigned” to each camera.

For each camera, there is a parameter called Bandwidth Assigned. This read only parameter indicates the total bandwidth that has been assigned to the camera. The Bandwidth Assigned parameter includes both the bandwidth that can be used for image data transmission plus the bandwidth that is reserved for packet resents and camera control signals. To determine the “data bandwidth assigned,” you must subtract out the reserve.

You can use the formula below to determine the actual amount of assigned bandwidth that is available for data transmission. To use the formula, you need to know the current value of the BandwidthAssigned parameter and the BandwidthReserve parameter for each camera.

$$\text{Data Bandwidth Assigned} = \text{Bandwidth Assigned} \times \frac{100 - \text{Bandwidth Reserved}}{100}$$

Step 6 - For each camera, compare the data bandwidth needed with the data bandwidth assigned.

For each camera, you should now compare the data bandwidth assigned to the camera (as determined in step 4) with the bandwidth needed by the camera (as determined in step 3).

For bandwidth to be used most efficiently, the data bandwidth assigned to a camera should be equal to or just slightly greater than the data bandwidth needed by the camera. If you find that this is the situation for all of the cameras on the network, you can go on to step 6 now. If you find a camera that has much more data bandwidth assigned than it needs, you should make an adjustment.

To lower the amount of data bandwidth assigned, you must adjust a parameter called the Inter-PacketDelay. If you increase the Inter-PacketDelay parameter value on a camera, the data bandwidth assigned to the camera decreases. So for any camera where you find that the data bandwidth assigned is much greater then the data bandwidth needed, you should do this:

1. Raise the setting for the Inter-packetDelay parameter for the camera.
2. Recalculate the data bandwidth assigned to the camera.
3. Compare the new data bandwidth assigned to the data bandwidth needed.
4. Repeat 1, 2, and 3 until the data bandwidth assigned is equal to or just greater than the data bandwidth needed.

If you increase the inter-packet delay to lower a camera's data output rate there is something that you must keep in mind. When you lower the data output rate, you increase the amount of time that the camera needs to transmit an acquired frame (image). Increasing the frame transmission time can restrict the camera's maximum allowed frame rate.

Step 7 - Check that the total bandwidth assigned is less than the network capacity.

1. For each camera, determine the current value of the BandwidthAssigned parameter. The value is in Byte/s. (Make sure that you determine the value of the BandwidthAssigned parameter after you have made any adjustments described in the earlier steps.)
2. Find the sum of the current BandwidthAssigned parameter values for all of the cameras.

If the sum of the Bandwidth Assigned values is less than 125 MByte/s for a GigE network or 12.5 MByte/s for a 100 Bit/s network, the bandwidth management is OK.

If the sum of the Bandwidth Assigned values is greater than 125 MByte/s for a GigE network or 12.5 MByte/s for a 100 Bit/s network, the cameras need more bandwidth than is available and you must make adjustments. In essence, you must lower the data bandwidth needed by one or more of the cameras and then adjust the data bandwidths assigned so that they reflect the lower bandwidth needs.

You can lower the data bandwidth needed by a camera either by lowering its frame rate or by decreasing the size of the area of interest (AOI). Once you have adjusted the frame rates and/or AOI settings on the cameras, you should repeat steps 2 through 6.

Revision History

Doc. ID Number	Date	Changes
AW144501000	10 Jul 2017	Initial release of this document.