WHITE PAPER

Advantages of Using TCP for GigE Vision Devices

V1.0.1
2023-May-12

Scope

The GigE Vision Standard is a mature machine vision protocol standard based on UDP. The original protocol is designed for low latency and low overhead to support high throughput on Gigabit Ethernet infrastructure.

As Ethernet technology advances and throughput demands grow, some limitations of the GigE Vision protocol got more apparent over time. In this paper, we show a method to improve the throughput and reliability of the system by utilizing commonly available hardware to offload features of the network adapter on the host.

The application example is based on the new Allied Vision Alvium G5X camera series, which is a GigE Vision compliant product with a 5GBASE-T interface.

Introduction to GigE Vision

GigE Vision is a widely implemented standard in the machine vision industry. Many products are available, such as different kinds of devices and host software. Cameras with area image sensors are the most common devices, but there are other types of sensors available that may demand high data throughput as well.

The standard defines two main protocols with specific functions, the GigE Vision Control Protocol (GVCP) and the GigE Vision Streaming Protocol (GVSP). Both use UDP datagrams for data transfer. UDP has the advantage of having a low overhead on the wire and providing minimal transmission latency. But it has also drawbacks: The order of datagrams arriving at the host may be not the same as the sent order. Also, UDP does not support any means to detect completeness of transmitted data, each datagram stands for its own.

To reach the required reliability, the GigE Vision standard adds different mechanisms on top of UDP to maintain the order and completeness of transmission. For example, the GVCP allows only one command at a time, and it supports an acknowledge message for each command. In combination with a timeout, the controlling host can detect data loss, retry, or decide what to do otherwise.

The control channel via GVCP is usually not an issue regarding throughput, because the required bandwidth to control a device like a camera usually is very low. So let's look at the streaming channel for the payload data, which is based on GVSP. This protocol works differently: Each GVSP packet header contains an image ID, a packet ID, and some payload information. By inspecting this information, the host can check completeness and reconstruct the correct order of the payload data. If a packet is missing, the host may request to resend the data from the device after a certain timeout, provided that the device supports this.

The issue with the processing overhead

Each network packet that arrives at the host's network interface card (NIC) causes a certain processing overhead. For example, the NIC activates an interrupt to notify the device driver. The packet is handed over by the driver to the network stack of the operating system to identify the receiving application. Then, in case of GVSP, the GigE Vision transport layer inspects the packet to identify the payload and to copy it to the frame buffer of the user application. In addition, the transport layer checks the completeness and correct order of the packets as described above.

All this processing is usually implemented in software, which means each packet is touched by a CPU probably multiple times. Often the overhead time spent per packet is a dominating factor in comparison to the time spent to process the actual payload of the packet, see Figure 1. As the throughput of network interfaces has been growing over the years, but not the packet size accordingly, the load for the host system to process the specific amount of payload is still approximately the same.

Figure 1: CPU overhead is relatively independent of packet size.

There are many technical ways to reduce this burden. Optimizations in driver and operating system, like maximizing the packet size, increasing the number of receive buffers, activating interrupt moderation, or bypassing Microsoft Windows' network stack via a filter driver are usually easy to achieve, because these are quite standard features on every system. For throughput rates beyond typically a few Gbit/s, these optimizations are mandatory to get a reliable transmission. But there often are bottlenecks which are hard to remove. The UDP packet throughput is often limited by the single-thread performance of a system. Certain resources, like NIC interrupt handling, sometimes do not scale well on multi-core systems.

To improve things, hardware offloading comes into play. There are different solutions on the market, ranging from standard network adapters with custom configuration of their offloading engine, to custom GigE Vision frame grabber hardware. The latter usually writes the assembled image payload directly to the application's frame buffer via DMA, without the need of the CPU to touch any GVSP packet. This frees up lots of CPU cycles for other tasks, but obviously adds cost to the system for the custom hardware and reduces the flexibility in the selection of system components and software.

Why TCP?

Now we step a little bit back from the full GVSP hardware offloading to a less custom and more flexible solution: The nearly omnipresent offloading features of standard NICs.

As TCP is a widely used protocol, many years of optimization knowhow have been implemented in the products. TCP certainly has several properties that might be interesting to be discussed for machine vision applications. In this document we focus on one specific, quite common performance optimization feature available for TCP, which is called large receive offload (LRO). Microsoft calls it receive side coalescing (RSC).

Using TCP for the transmission of GVSP payload seems contradictory at first glance. GVSP is designed for maximum throughput and minimal latency. The TCP protocol does not have this as its primary goal, its focus is to guarantee the completeness and order of the data.

Also, the TCP header is typically 12 bytes bigger than the UDP header, which causes slightly more transmission overhead. But using LRO in the system can overcompensate the disadvantages.

The basic idea behind LRO is that the NIC collects multiple packets received from a TCP connection and re-packs them autonomously to one bigger TCP packet. Then, this big TCP packet is handed over to the operating system, as if it has been transmitted by the far end with that size. Figure 2 shows the process. This mechanism allows to support packet sizes that are even bigger than the MTU size of the network hardware and can reach up to 64 kiB. LRO is transparent, the existing components in the system are not required to be aware of it.

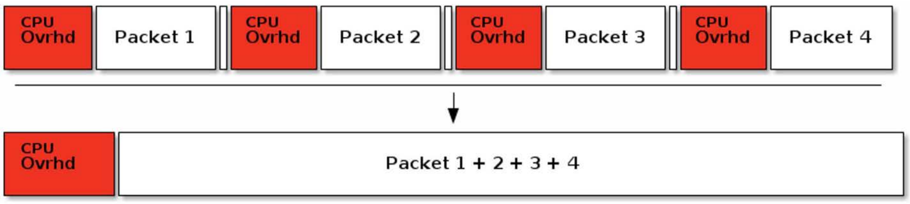

Figure 2: LRO combines multiple received TCP packets to bigger packets on the NIC hardware, before handing it over to the software.

Enabling LRO on the network adapter can reduce the number of packets that need to be processed by the host network stack to a fraction. The number of interrupts and system calls decreases significantly. This frees up system resources for the user application.

Packet size versus transfer size

As GigE Vision is based on UDP, the standard connects the concept of GVSP packets with UDP datagrams. Practically, this means that each logical GVSP packet must exactly fit into one UDP datagram. The maximum GVSP packet size is limited to the maximum UDP datagram size that itself is limited to the maximum transmission unit (MTU) size of the network path. While the device transmits a stream via GVSP, the host receives (UDP) packet by (UDP) packet, decodes each GVSP header and copies the GVSP payload to the frame buffer.

To minimize system load, Jumbo Frames – Ethernet frames larger than 1518 bytes – are usually enabled on all network devices. A typical supported size for a Jumbo Frame is 9 or 16 kiB.

On the other hand, when streaming GVSP data via a TCP connection, the data is transferred to the host application as a sequence of bytes. The TCP protocol ensures segmentation, retransmission, and flow control. The transmitting device as well as the receiving application do not need to take care about datagrams or packets anymore. This is completely covered by the underlying TCP implementation.

However, the concept of a logical GVSP packet, or let's call it GVSP transfer to distinguish it here, is useful in the TCP use case as well. It allows big images to be split into smaller pieces, which can help to balance trade offs in system components, such as buffer resources or latency. Figure 3 on page 4 shows an example of how multiple GVSP transfers might be packaged into independently sized TCP packets.

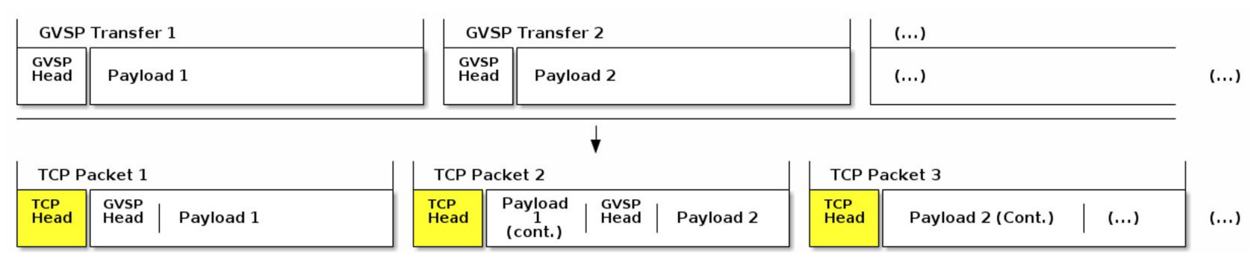

Figure 3: When using TCP, the concept of GVSP transfers allows independence between GVSP transfer size and TCP packet size.

We have two separate layers – GVSP transfers on top of TCP packets – and each has its own optimizations. TCP can be optimized via LRO as described above. On the other hand, decoding the GVSP transfers creates independent processing overhead, which means that increasing the size of GVSP transfers helps to further reduce the system load.

Other aspects of performance and system design

Throughput in traditional GigE Vision (UDP) manner is controlled by setting a predefined throughput limit on the device side: By limiting the acquisition frame rate and / or by setting an inter-packet delay before the streaming is started. The system must ensure that the allocated bandwidth is always available. This tends to promote oversized systems, because a certain margin is needed to cover short-term overload situations and get the required reliability with UDP.

The low-level Ethernet flow control (defined in the IEEE 802.3 standard) helps here. It reduces the required margin in the system a lot: Network links can be nearly saturated and a UDP transmission gets much more reliable. However, this requires an appropriately sized buffer in the device to absorb all back pressure caused by Ethernet flow control. But this buffer usually is available in a camera to serve GigE Vision resend requests anyway. It is much more effective to pause the transmission for a short time instead of losing a packet, waiting for timeout, sending a resend request, and transmitting the packet again.

TCP benefits as well from the Ethernet flow control mechanism because this avoids loss of TCP packets due to buffer overflows in network devices. But additionally, it has its own flow control and congestion control mechanisms on top of that.

Implementation of a camera device

Allied Vision is introducing the new camera series Alvium G5X as a variant of the Alvium G5 GigE Vision camera, which adds the capability of streaming GVSP over TCP. The camera has a 5GBASE-T network port. It is fully backwards compatible to the Alvium G5 and the current GigE Vision Standard version 2.2.

This camera integrates a TCP protocol implementation in an FPGA that is based on the Network Protocol Acceleration Platform (NPAP) by Missing Link Electronics (MLE). MLE's NPAP offers a compact UDP/TCP/IP full accelerator which can be customized to fit in many embedded applications. For example, with an asymmetric use case like cameras, where one transfer direction is heavily utilized, but the opposite direction is only used for control commands, the buffer size can also be set up asymmetrically. This helps to save precious resources within the FPGA. Furthermore, the number of simultaneous TCP sessions can be scaled in the design phase. This is important as each TCP session is represented in logic. Lastly, NPAP is adjustable in speed and allows high speed communications even for small devices. The 128-bit bus within NPAP allows 10GigE with a very low clock speed of 78.125 MHz.

Performance test on an example system

To demonstrate the effect of TCP versus UDP streaming, Allied Vision performed tests as described in the following. The simple setup consists of a 5-megapixel Alvium G5X-511c camera, connected via a 10-Gbit/s-class switch to an embedded host system that controls the camera and receives the stream data, as shown in Figure 4.

The embedded host is an NVIDIA Jetson Xavier NX module mounted on a STEMMER IMAGING Modular Embedded Board. This carrier board provides dual 10GBASE-T network connectivity based on an Intel x550 chipset. For more information, see https://help.commonvisionblox.com/NextGen/14.0/md_theory_of_operation_embedded__embedded.html.

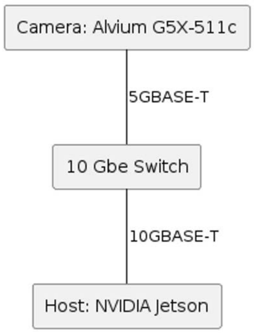

Figure 4: Test setup

The host runs a Linux operating system. Network device settings are optimized for TCP and UDP. This includes especially activating the LRO feature of the NIC and Jumbo Frame support (MTU size of 9000 bytes in this case). Also, the network receive buffer size has been increased.

GenICamBrowser, a part of the STEMMER IMAGING Common Vision Blox (CVB) software package, is used as the host application. It controls the camera, receives, and displays the images. It further provides statistics about the GVSP connection.

In UDP mode, the camera was configured to a GVSP packet size that utilizes the full path MTU size of 9000 bytes. Each UDP packet carries one complete GVSP packet. On the other hand, in TCP mode, the GVSP transfers are not constrained by the MTU. The camera was configured to a larger size of 63000 Bytes accordingly.

The goal is to stream images from the camera at full camera speed (approximately 4.3 Gbit/s) without losing any images. The number of dropped frames is not the only metric used to evaluate the performance. Additionally, we determine the average CPU utilization and the maximum single core utilization of the host.

Test results

Using UDP to stream the GVSP data is not stable in this example system, as shown in Figure 5 on page 6. Every one or two minutes an image gets lost, resulting in a loss ratio of more than 0.1%. The most loaded CPU seems to be the bottleneck: It is around 95% of the time busy, with peaks to 100% (corresponding to a load value of up to 1.0). An additional load that is occurring sporadically, may lead to overload situations where packets are missed and resend requests can't be handled in time to recover the images.

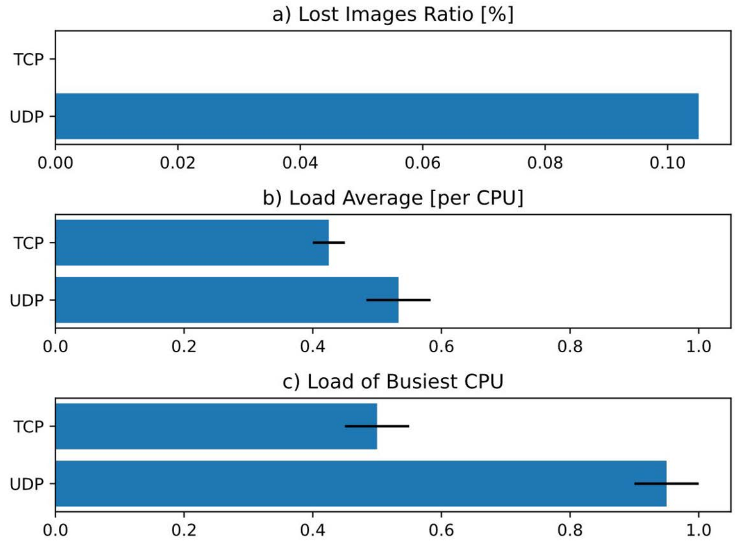

Figure 5: Summary of test results showing reliability and CPU load comparison between UDP (GVSP packet size 9000 bytes) and TCP (GVSP transfer size 63000 bytes, LRO enabled)

The TCP based GVSP stream, in comparison, did not lose any images. The busiest CPU with a load value around 0.5 is only slightly more busy than the average of all CPUs, so there seems to be no critical bottleneck anymore. The average load is approximately 20% lower, leaving more CPU time for the actual application. Refer to Figure 6 for a screenshot of this scenario.

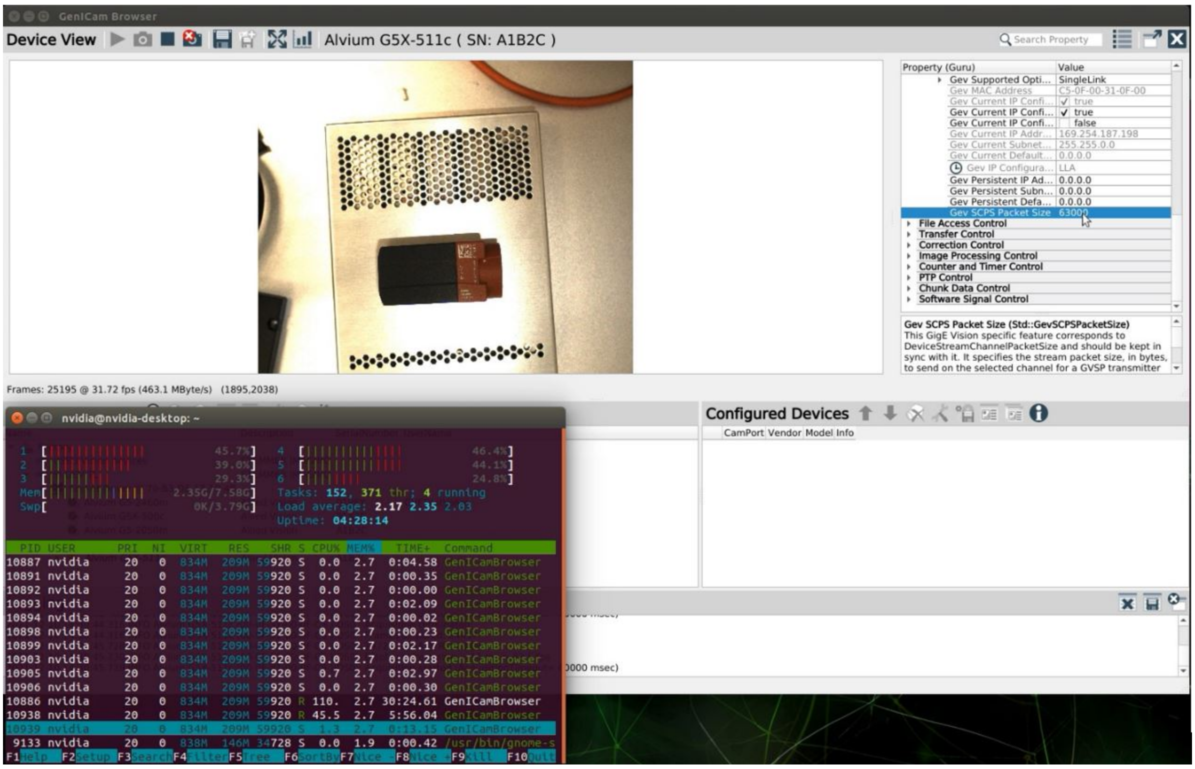

Figure 6: Example screenshot from test run streaming via TCP. Note that the camera prototype uses the feature Gev SCPS Packet Size to control both: GVSP packet size in UDP mode and GVSP transfer size in TCP mode. This may change in future releases.

Conclusion and outlook

It has been demonstrated that a TCP transmission using large receive offload can reduce the system load significantly in comparison to UDP based GVSP. In our example system with a simple image display application, the CPU load could be reduced by 20%. This solution can be implemented without additional hardware cost on the host system. The efficiency improvement enables reliable applications even on resource limited embedded systems, where classical GVSP via UDP would not be reliable enough.

This approach can be interesting for small or medium sized machine vision systems with high throughputs. With TCP, higher cost of a full hardware offloading solution or a more powerful host can be avoided.

With the introduction of the Alvium G5X series, Allied Vision is proud to pioneer TCP support in 5GigE cameras within the Machine Vision industry. This greatly improves usability for high-throughput, embedded, and multi-camera systems. The Alvium G5X series delivers enhanced reliability while relieving host systems by optimized packed handling through suitable network interface cards.

Authors and contributors

This document was created by Jörg Preckwinkel, Firmware Developer, Allied Vision Technologies GmbH https://www.alliedvision.com/en/. Many thanks for their support and contribution go to Andreas Schuler, Director Applications, Missing Link Electronics https://www.missinglinkelectronics.com and to Andreas Rittinger, Team Manager Image Acquisition Software Development, STEMMER IMAGING AG https://www.stemmer-imaging.com/en/.

Allied Vision Technologies GmbH https://www.alliedvision.com/en/

For more than 30 years, Allied Vision has been helping people to reach their imaging goals. Allied Vision supplies camera technology and image capture solutions for industrial inspection, medical and scientific imaging, traffic monitoring, and many more application areas in digital imaging.

With a deep understanding of customers' needs, Allied Vision finds individual solutions for every application, a practice which has made Allied Vision one of the leading camera manufacturers worldwide in the machine vision market. The company has nine locations in Germany, Canada, the United States, Singapore, and China and is represented by a network of sales partners in over 30 countries.

Missing Link Electronics https://www.missinglinkelectronics.com

Missing Link Electronics is a Silicon Valley based technology company with offices in Germany. MLE is a partner of leading electronic device and solution providers and has been enabling key innovators in the automotive, industrial, test, and measurement markets to build better embedded systems very fast.

MLE's mission is to develop and to market technology solutions for embedded systems realization via pre-validated IP and expert application support, and to combine off-the-shelf FPGA devices with open-source software for dependable, configurable embedded system platforms.

Missing Link Electronics' expertise is covering domain-specific architectures, I/O connectivity, and acceleration of data communication protocols, additionally opening up FPGA technology for analog applications, and the integration and optimization of open-source Linux and Android software stacks on modern extensible processing architectures.

STEMMER IMAGING AG https://www.stemmer-imaging.com/en/

STEMMER IMAGING is the leading international partner for machine vision technology. Their portfolio for both industrial and non-industrial applications combines an extensive commercial range of products with a high level of expertise and value-added services. In addition, STEMMER IMAGING develops subsystems to solve specific tasks. For over 35 years, STEMMER IMAGING has been helping their customers to take a leading role in their markets – across Europe and Latin America.

Contact us

Website, email

General
www.alliedvision.com/en/contact
info@alliedvision.com

Distribution partners
www.alliedvision.com/en/avt-locations/avt-distributors

Support
www.alliedvision.com/en/support
www.alliedvision.com/en/about-us/contact-us/technical-support-repair-/-rma

Offices

Europe, Middle East, and Africa (Headquarters)

Allied Vision Technologies GmbH
Taschenweg 2a
07646 Stadtroda, Germany
T// +49 36428 677-0 (Reception)
T// +49 36428 677-230 (Sales)
F// +49 36428 677-28

Asia-Pacific

China

Allied Vision Technologies
(Shanghai) Co., Ltd.
2-2109 Hongwell Int. Plaza
1602# ZhongShanXi Road
Shanghai 200235, China
T// +86 21 64861133

Singapore

Allied Vision Technologies Asia Pte. Ltd
82 Playfair Rd, #07-02 D'Lithium
Singapore 368001
T// +65 6634 9027

North, Central, and South America

Canada

Allied Vision Technologies Canada Inc.
300 – 4621 Canada Way
Burnaby, BC V5G 4X8, Canada
T// +1 604 875 8855

USA

Allied Vision Technologies, Inc.
102 Pickering Way- Suite 502
Exton, PA 19341, USA
Toll-free// +1-877-USA-1394
T// +1 978 225 2030

Copyright and trademarks