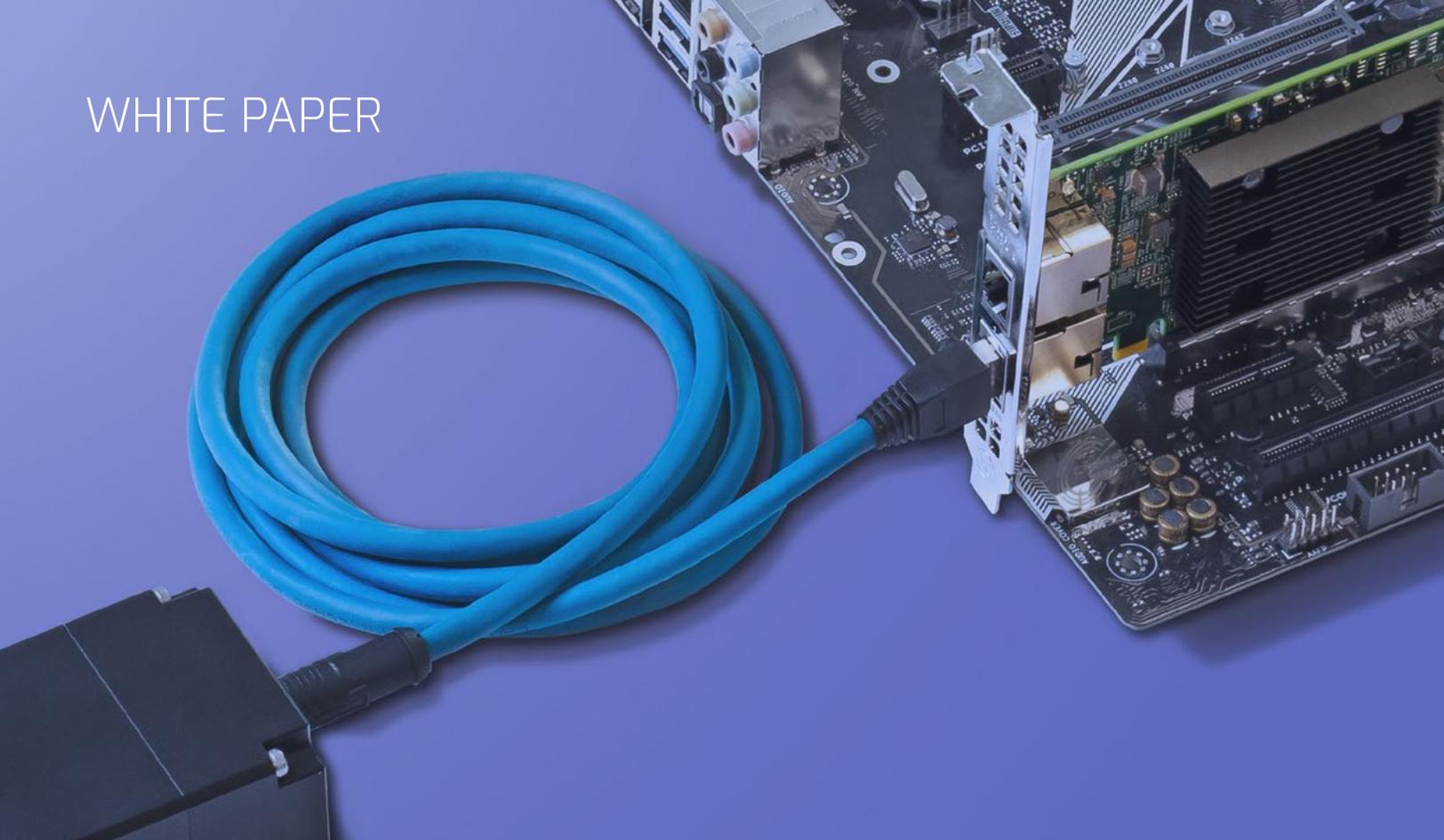


WHITE PAPER



# High-Speed, Maximum Reliability: REMOTE DIRECT MEMORY ACCESS WITH ATLAS10 CAMERAS

The landscape of high-speed data transmission is in a constant state of evolution, as new technologies emerge to meet the ever-growing demands of bandwidth-intensive applications. Ethernet UDP protocol has been the preferred convention for streaming data from GigE Vision cameras for many years due to its simplicity and low latency. However, as the demand for bandwidth applications beyond 1 GigE continues to rise, a more efficient and reliable transmission standard is necessary to handle the massive increase in data. This has resulted in the adoption of Remote Direct Memory Access, or RDMA, as a viable alternative to UDP for 10GigE and 25GigE multi-camera applications.

While UDP offers exceptional performance and simplicity, it has inherent limitations when it comes to flow control and packet retransmissions. RDMA, on the other hand, provides a more robust and efficient data transmission method by bypassing the CPU and operating system to store image data directly onto the host PC's memory. This frees up the CPU for

## WHAT'S INSIDE:

[Why Use UDP for the GigE Vision standard?](#)

[Challenges of UDP for 10GigE Cameras](#)

[Removing the CPU Bottleneck](#)

[Remote DMA over Convergent Ethernet \(RoCE v2\)](#)

[RDMA Connection and Transfer Steps + Notes](#)

[RDMA for GigE Vision](#)

[What about GPU Memory?](#)

[UDP versus RDMA Benchmarks](#)

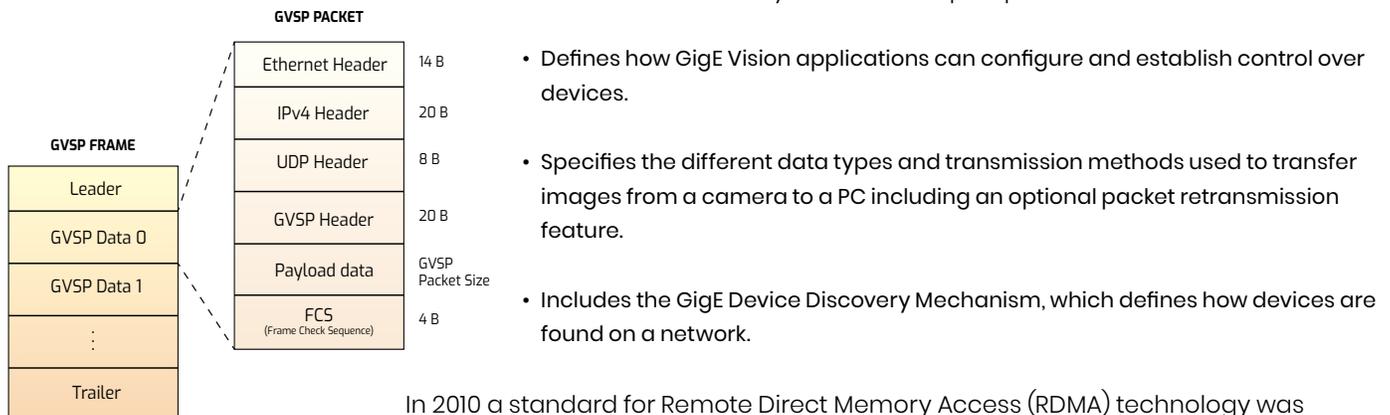
[The Future is Fast](#)

other critical vision processing tasks, resulting in faster and more efficient image streams. Moreover, RDMA's built-in reliability features, including flow control and packet retransmission, make it the ideal choice for managing the massive amounts of data produced by modern high-bandwidth applications.

If you are seeking a more dependable, high-speed data transmission solution for your GigE Vision cameras, it is time to take a closer look at RDMA. In this article, we delve into how this technology can optimize your vision applications and offer new insights into its benefits over traditional Ethernet protocols.

## WHY USE UDP FOR THE GIGE VISION STANDARD?

When the GigE Vision Control Protocol (GVCP) was established (2006), camera vendors chose UDP as the networking protocol not only for its efficient streaming performance, but also for its simplicity. Operators, designers, and integrators could quickly implement UDP into the standard and add missing reliability features on top of UDP, with those features running at the application level in software, utilizing CPU resources on the host PC and the camera's firmware. These reliability features are optional for camera vendors making it even simpler if vendors choose to forgo these features. The GigE Vision standard enables Ethernet devices to communicate using UDP by defining device control, stream, discovery, and the transport protocols over Ethernet. The GVCP:



**figure 1:**  
Breakdown of a GVSP Frame and Packet

In 2010 a standard for Remote Direct Memory Access (RDMA) technology was introduced and named Remote DMA over Convergent Ethernet (RoCE, pronounced "Rocky") RoCE v1. But it was not until the creation of the RoCE v2 standard in 2014 that RDMA technology became more versatile, with the ability to function over Layer 3 (IP) networks. (Further details on this topic can be found on page 4). At the time of its release, RDMA technology was intended primarily for use in data center environments and lacked readily available consumer-grade RDMA hardware and components. The degree of hardware and software vendor support also differed significantly between the GigE Vision and RoCE standards, with GigE Vision being designed and supported by machine vision camera and software manufacturers, and RoCE receiving support primarily from IT and semiconductor corporations.

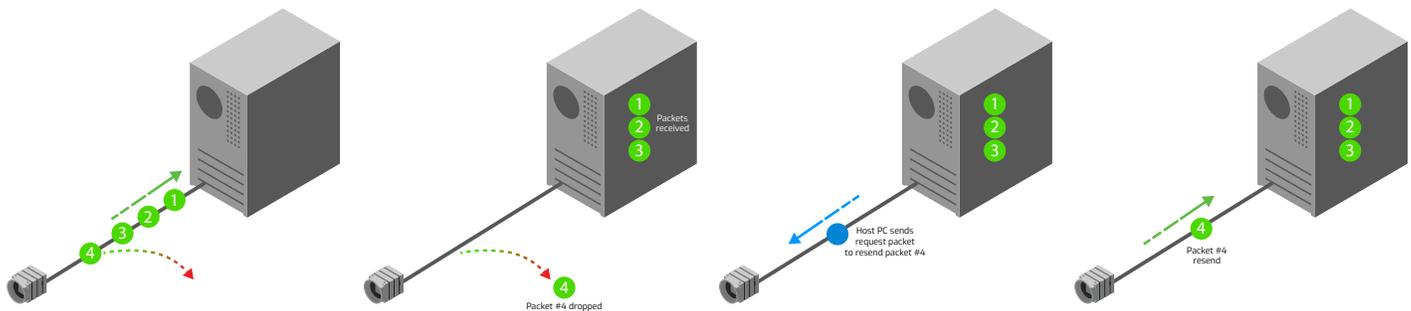


**figure 2:** GigE Vision logo  
The Automated Imaging Association (AIA) introduced GigE Vision in 2006. The standard works to unify integration and communication protocols over Ethernet between cameras, hardware components, and software packages.

Even though UDP was the chosen protocol, camera manufacturers and customers understood that data transfer reliability could not be completely ignored. The GigE Vision standard includes sections on reliability features similar to Transmission Control Protocol (TCP), and these optional features are added to the header of each data packet. The GigE Vision standard includes a GVSP header with sequence information allowing packets of a frame to be sent out-of-order and realigned after delivery. The GVSP header also facilitates packet retransmission if a packet is dropped.

# CHALLENGES OF UDP FOR 10GIGE CAMERAS

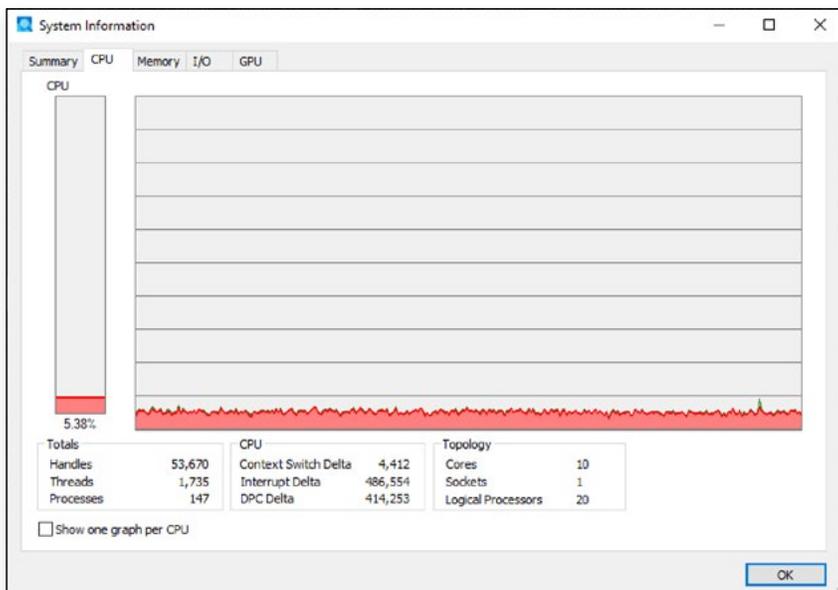
The current implementation of UDP for GigE Vision was used with 1 GigE bandwidth in mind. As mentioned earlier, reliability features missing from UDP are built into the GigE Vision standard at the application layer. As a result, any reliability feature that is used will be limited by the amount of available CPU resources on the host, the quality of the camera's software and filter driver, and how robust the camera's firmware is at handling extra requests. For example, for many camera manufacturers, packet retransmission in GigE Vision requires a filter driver — a specialized software driver on the host PC — to monitor for any missing packets. When the PC notices a missing packet, it sends a retransmission request to the camera. The camera then needs to parse the packet, which usually takes place in the camera's firmware, and then fetch that packet from the camera's image buffer. The camera must interrupt normal transmission and retransmit the requested packet. If several packets are missing, interspersed throughout the frame, both the PC and camera are burdened. However, at 1 GigE speeds, CPU and camera resources needed to monitor and resend dropped packets are readily available.



**figure 3: Packet Retransmission**

Example of the host PC dealing with dropped packets. Because UDP doesn't offer any built-in reliability features, data delivery is monitored by the filter driver supplied by the camera manufacturer. This driver reads the GVSP header on each packet.

In addition, camera manufacturers have had many years to develop and fine-tune their filter drivers. Filter drivers not only allows 1 GigE cameras to operate effectively while maintaining reliability, but also with very little CPU utilization. Even at speeds above 1 GigE, LUCID has worked to optimize the UDP data transfer under GigE Vision. Our filter driver provides fast, low latency, and optimized CPU performance, along with all the optional reliability features available to users. For the more efficient UDP performance, installing LUCID's filter driver (LUCIDLwf.sys) provides fast packet monitoring with image assembly. For example, streaming four Atlas10 10GigE cameras simultaneously at full bandwidth results in only an average of 5.4% CPU resources.



**figure 4: Atlas10 Using UDP Filter Driver**

Running multi-10GigE Atlas10 cameras with optimized CPU resources using LUCID's UDP filter driver.

See more benchmark results on page 10.

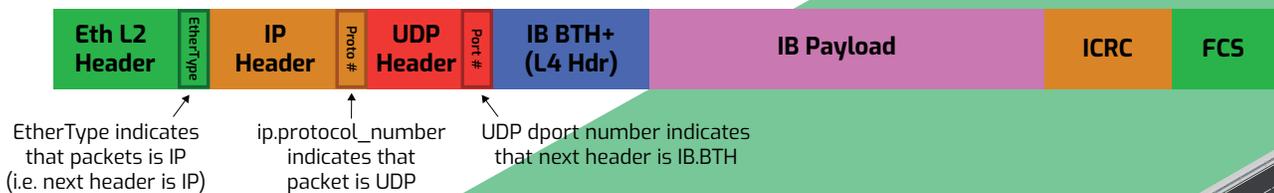
# REMOVING THE CPU BOTTLENECK

Requirements for high throughput and low latency are not unique to industrial machine vision and have been a common issue for high performance computing (HPC) for many years. Distributed computing clusters used in cloud computing or machine learning applications must deal with a vast amount of data transactions on a large scale and in turn generate an exponential number of input and output (I/O) operations. I/O, the process of sending and receiving data for processing, has historically been controlled by the CPU. In a conventional networking stack, received packets are stored in the operating system's memory (kernel space) and then copied to the application memory (user space). This consumes CPU cycles and introduces latencies. The issue is compounded when multiple network interfaces are used, which increases the number of interrupts and context switches required to handle the flow of incoming data. In these cases, the CPU might become the bottleneck limiting overall system performance. Part of the problem can be solved by deploying new generations of CPU that provide faster cycles and expanded memory capabilities. The optimum solution, however, is more than just faster CPU and raw bandwidth.

RDMA enables the movement of data between devices on a network with no CPU involvement on per packet basis. Network adapters that implement RDMA enable writing data directly into application memory, bypassing the operating system entirely, avoiding unnecessary data copy (zero-copy functionality), and therefore reducing CPU overhead.

## REMOTE DMA OVER CONVERGENT ETHERNET (RoCE V2)

The first implementation of RDMA protocol was realized over InfiniBand, a high-performance and low-latency network fabric used for data interconnect among and within computers. InfiniBand rapidly gained popularity for use in high-performance computing applications. Today, most of the HPC systems on the TOP500 list of the most powerful commercially available computer systems use RDMA. Its broader adoption, however, was hampered by the need to procure dedicated InfiniBand switches, which do not leverage the extensive Ethernet infrastructure already in place. To answer this need, RoCE v1 was launched in 2010 as an open standard supported by the InfiniBand Trade Association (IBTA). The RoCE specification was expanded with the RoCE v2 update released in 2014 to provide routing across Layer 3 networks (IP) and flow control capabilities.

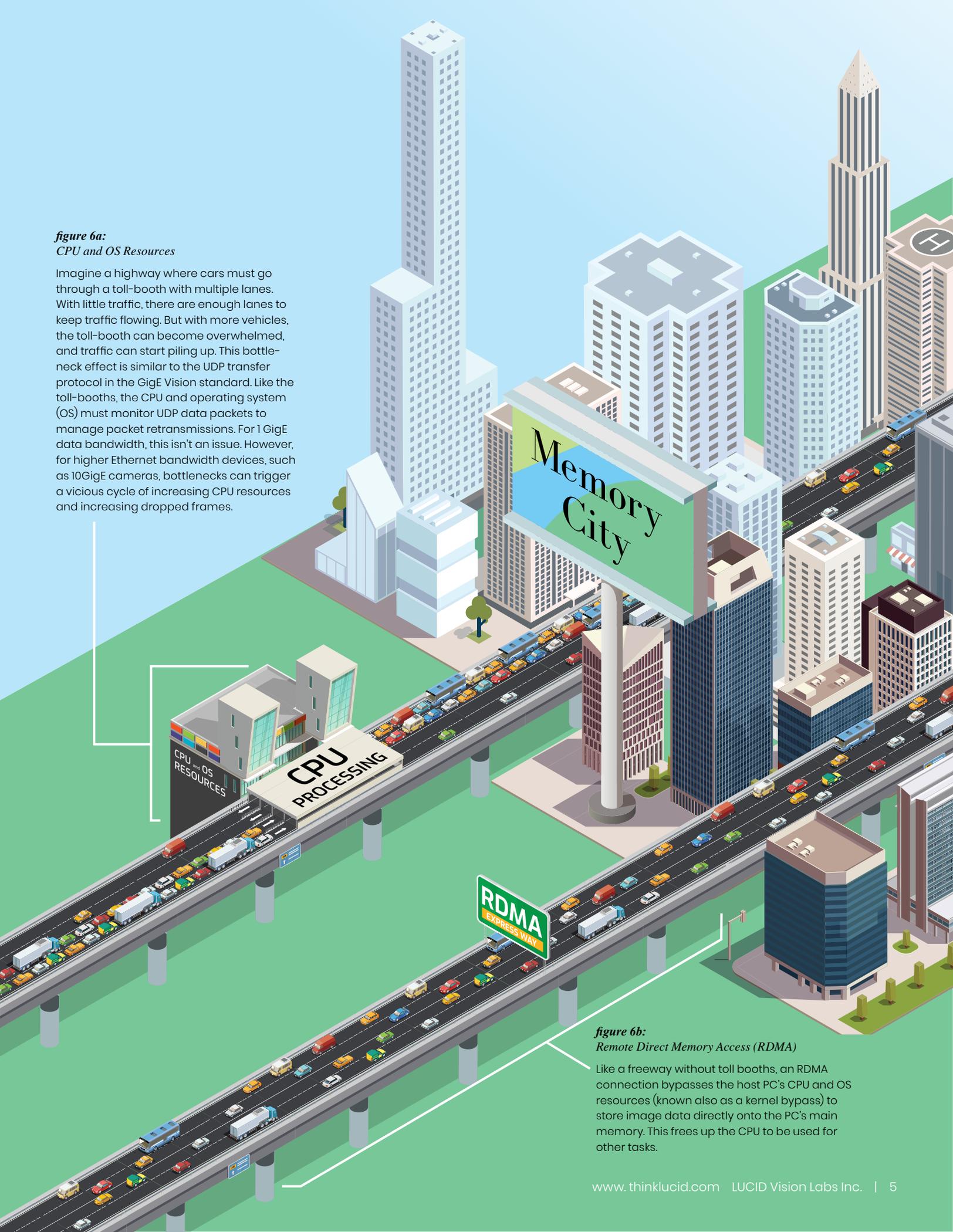


**figure 5: RoCE v2 Packet Format**

The RoCE v2 packet is encapsulated on top of either the UDP/IPv4 or the UDP/IPv6 protocol making it routable across multiple subnets. Therefore, RoCE v2 is sometime called Routable RoCE (RRoCE).

**figure 6a:**  
*CPU and OS Resources*

Imagine a highway where cars must go through a toll-booth with multiple lanes. With little traffic, there are enough lanes to keep traffic flowing. But with more vehicles, the toll-booth can become overwhelmed, and traffic can start piling up. This bottleneck effect is similar to the UDP transfer protocol in the GigE Vision standard. Like the toll-booths, the CPU and operating system (OS) must monitor UDP data packets to manage packet retransmissions. For 1 GigE data bandwidth, this isn't an issue. However, for higher Ethernet bandwidth devices, such as 10GigE cameras, bottlenecks can trigger a vicious cycle of increasing CPU resources and increasing dropped frames.



**figure 6b:**  
*Remote Direct Memory Access (RDMA)*

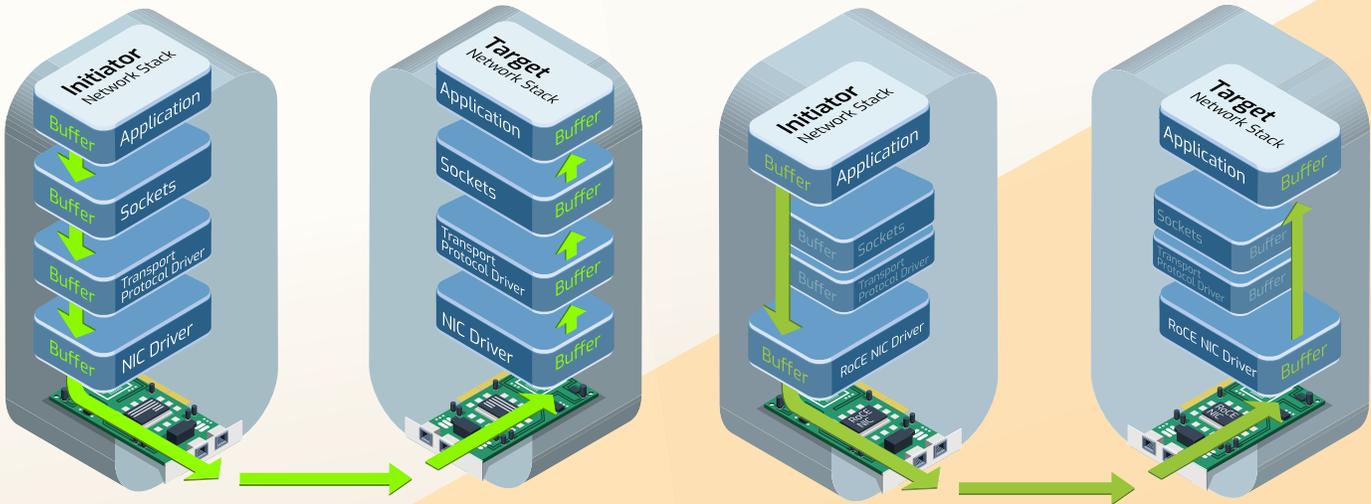
Like a freeway without toll booths, an RDMA connection bypasses the host PC's CPU and OS resources (known also as a kernel bypass) to store image data directly onto the PC's main memory. This frees up the CPU to be used for other tasks.

Adapting RDMA over the Ethernet network enables the following benefits:

- Fastest throughput and lowest latency available at all speeds of Ethernet networks, compatible with existing switching infrastructure and cabling.
- Complete hardware offload of packet handling with no CPU involvement through zero-copy; ability to send and receive data to and from remote buffers.
- Comprehensive ecosystem of industrial connectivity solutions offering secure connectors, EMI shielding, ground isolation, and Power over Ethernet (PoE).
- Supported by many vendors of both hardware and software solutions – including Broadcom, Marvell, Nvidia, and Intel – promoting interoperability.

**What do you need to enable RDMA with LUCID 10GigE Cameras?**

- LUCID Atlas10 camera with RDMA firmware
- RDMA 10GigE Host Channel Adapter (HCA)
- (Optional) 10GigE Switch with PFC priority-enabled VLAN
- Cat6 or better Ethernet cable



**figure 7:** In a conventional networking stack (left), data transfer from one application to another application on a remote machine involves multiple buffer copies and context switching to mobilize the CPU at each stage. In an RoCE network, data is transferred directly from the initiator application to the target application with no CPU involvement, bypassing the OS stack entirely.

The Host Channel Adapter (HCA) manages RoCE operations, implementing in hardware all the logic needed to execute RDMA protocol. Data segmentation and reassembly as well as flow control are managed by the HCA allowing the sender and receiver applications to work with whole buffers. The RDMA channel is initiated by “pinning” memory. A memory region is reserved on the host for RDMA usage, the necessary protections are applied, and then the host passes the address to the HCA and removes itself from the data path. This registered memory region can now be used for any RDMA operation, bypassing the operating system, and generating no additional CPU load.

RoCE RDMA transactions use three queues. The send and receive queues handle all the data transaction and are always created together as a queue pair (QP). The completion queue (CQ) is used to track the completion of the work scheduled on the QP. The QPs enables application-level flow control to notify the sender of available buffers for RDMA transfer on the receiver’s end.

Before RDMA data transfers can take place, a queue pair (QP) and a completion queue (CQ) must be created for each hardware port, known as a channel adapter (CA) on the RDMA network.

- What is needed on the camera:
- Camera QP (Send and Receive Queues)**
  - Camera CQ (Completion Queue)**

Registering and reserving host memory for RDMA user applications is possible thanks to Network Direct in MS Windows® and the Libibverbs library in Linux.

RDMA data transfers can only start after memory is registered. This is done by "pinning" the host's memory. A memory region is reserved by the OS and registered with the HCA.

Host Channel Adapter (HCA) Port 2.  
HCA2 QP (Send and Receive Queues)  
HCA2 CQ (Completion Queue)

Host Channel Adapter (HCA) Port 1.  
HCA1 QP (Send and Receive Queues)  
HCA1 CQ (Completion Queue)

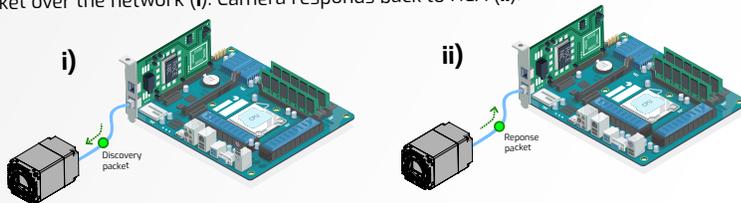
```
1  =====RDMA Verbs and Verbs APIs=====
2  //
3  //  How are RDMA devices programmed in applications? It starts with RDMA
4  // verbs, which are the low-level building blocks for RDMA applications. They
5  // are accessed by various API libraries, however, the two major libraries used
6  // are Libibverbs API for Linux and Network Direct SPI for Microsoft Windows.
7  // These APIs allow RDMA devices to establish channel adapter connections, pin
8  // and register memory, execute data transfers, and terminate connections.
9  //
10 //  There are two types of Verbs: one-sided and two-sided verbs. One-sided
11 // verbs allow remote devices (such as cameras) to completely bypass the CPU/OS
12 // when sending data. Two-sided verbs act more like traditional sockets that
13 // utilize the CPU/OS. LUCID uses two-sided verbs. Using two-sided verbs still
14 // removes several sources of CPU overhead compared to conventional Ethernet
15 // data transfers. Using two-sided verbs is necessary for requeuing transfers
16 // and polling the CQ. These tasks take up negligible CPU resources.
17 //
18 //
19 //
20 //
```

# RDMA CONNECTION & TRANSFER STEPS + Notes!

## STEP 1.

Discover devices on the RDMA network. RDMA HCA broadcasts discovery packet over the network (i). Camera responds back to HCA (ii).

This discovery process takes place over UDP and mimics how non-RDMA GigE Vision cameras discover each other.

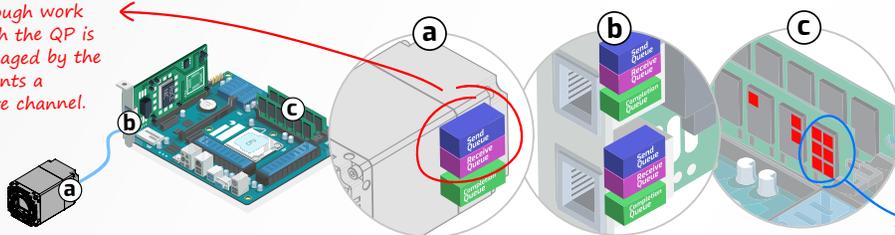


## STEP 2.

Generate a protected domain for the RDMA devices: Create **queue pairs (QP)** and **completion queues (CQ)** for all devices, **pin** host memory and **register** it with the RDMA HCA, and establish a **reliable connection (RC)** between the camera and HCA.

A queue pair (QP) consists of a send queue and a receive queue. These queues schedule the data transmission through work requests. Although the QP is created and managed by the drivers it represents a physical hardware channel.

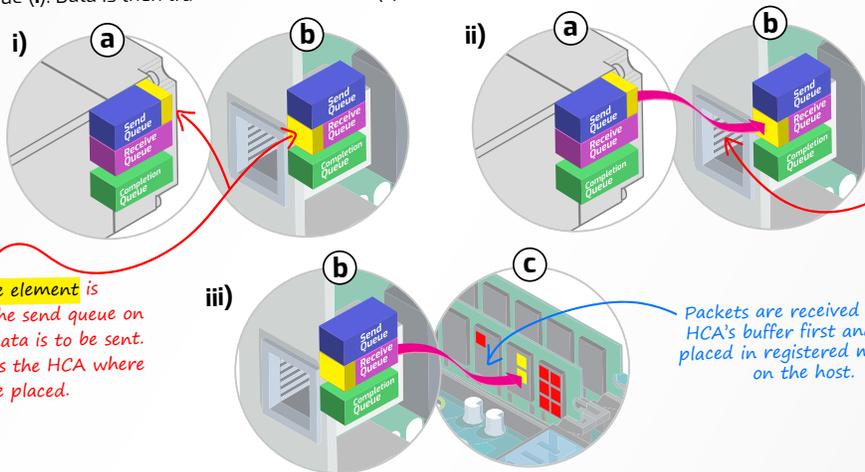
Reliable connection (RC) is one of 3 types of RDMA connections. The other two are unreliable connection (UC) and unreliable datagram (UD).



Pinning memory tells the OS to keep these sections resident and never have them paged. Registering memory is when pinned memory sections are registered with the RDMA HCA so it knows where to send the image data to.

## STEP 3.

Create a **work queue element (WQE)**, also known as **work request (WR)**, to initiate a data transfer. The work request is enqueued and adds a **send request (SR)** in the camera's send queue and a **receive request (RR)** in the HCA's receive queue (i). Data is then transferred to the HCA (ii) which then transfers the data to the host's memory (iii).



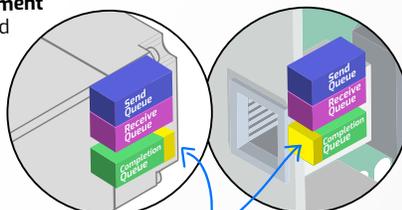
When a **work queue element** is processed, it tells the send queue on the camera what data is to be sent. On the HCA, it tells the HCA where that data should be placed.

During an RC data transfer, the HCA will send ACKs back to the camera. The ACKS enable packet retransmission similar to TCP.

Packets are received on the HCA's buffer first and then placed in registered memory on the host.

## STEP 4.

Add a **completion queue element (CQE)** to both the camera and the HCA's completion queue.



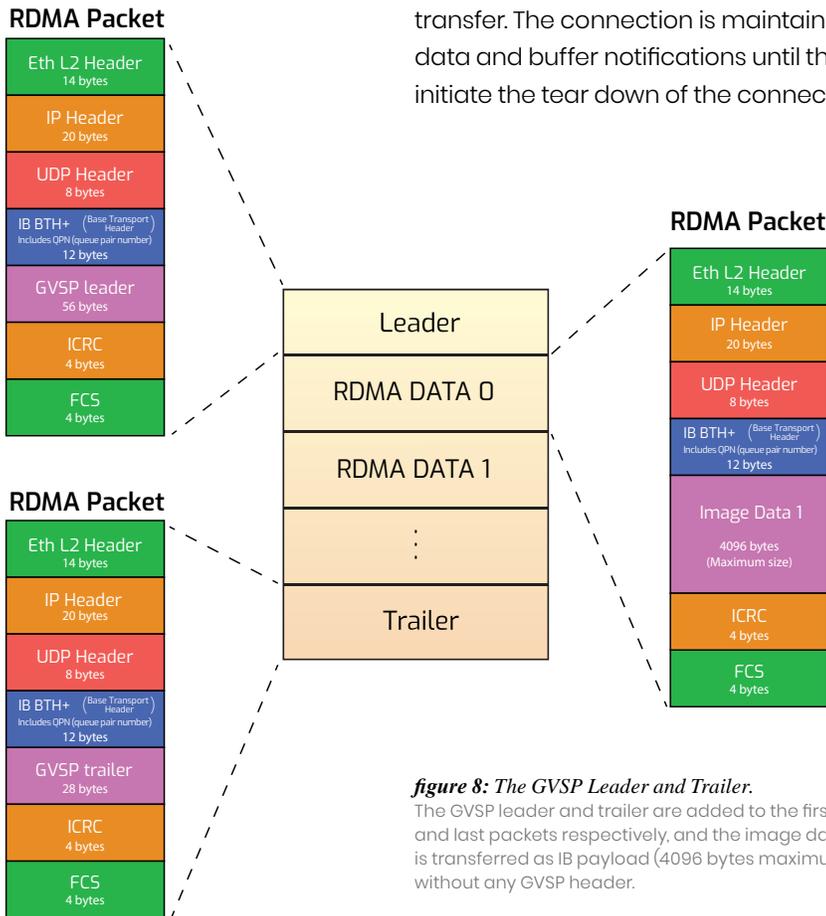
A **completion queue element** must be enqueued on both the camera and the HCA's completion queue before more work requests can begin.

## RDMA Acronym Cheat Sheet

- HCA = Host Channel Adapter (Similar to Network Interface Card - NIC)
- ACK = Acknowledgement Packet
- QP = Queue Pair
- SQ = Send Queue
- RQ = Receive Queue
- SR = Send Request
- RR = Receive Request
- RC = Reliable Connection
- WQE = Work Queue Element
- CQ = Completion Queue
- CQE = Completion Queue Element

# RDMA FOR GIGE VISION

Applications that transfer large blocks of data see the greatest efficiency gains from RDMA, making it well suited to benefit GigE Vision based cameras. To implement RDMA over GigE Vision, the GigE Vision Stream Protocol (GVSP) needs to be adapted. The receiver (host) establishes the RDMA stream channel by initiating a reliable connection (RC) with the device (the camera). One QP is created per stream channel with send and receive operations. The send queue (host → device) informs the device of available buffers on the receiver side while the receive queue (device → host) is used for payload transfer. The connection is maintained through the QP set up for the transfer of image data and buffer notifications until the receiver device sends a disconnect request to initiate the tear down of the connection.



**figure 8: The GVSP Leader and Trailer.**  
The GVSP leader and trailer are added to the first and last packets respectively, and the image data is transferred as IB payload (4096 bytes maximum) without any GVSP header.

Unlike GVSP using UDP, where GVSP headers are inserted into all data packets for a frame, GVSP using RDMA doesn't require any GVSP header for the image payload transfer. Since data payload contains no header, no decoding is necessary, and packets can be directly put into the contiguous memory destination buffer allocated on the receiver. Packet resends are handled by the RoCE v2 HCA making it unnecessary to track packet IDs through individual GVSP headers. RoCE v2 tracks packets with a packet sequence number (PSN) embedded in the InfiniBand Base Transport Header (IB BTH) and the receiver will only accept packets in order. If a gap in the PSN is detected (whether from a drop or from being out of order), the host will request re-transfer beginning with the packet after the last successfully received one. This reduces overhead and frees up CPU resources as all flow control and packet decoding is handled by the HCA running on dedicated hardware.

To implement flow control and provide packet retransmission capabilities, the camera must retain sent packets until the receiver has acknowledged ("ACK") their reception. Because it would be inefficient for the host to send an ACK for each packet it receives multiple packets can be acknowledged at once by sending an ACK for the latest in-order PSN that was received.

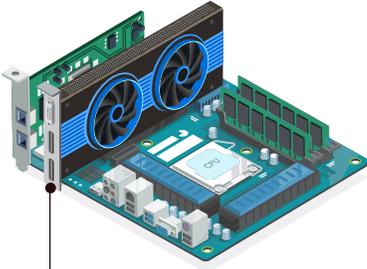
**figure 9: Profishark Screenshot of Packet Log from Atlas10 Camera to Host.**

Just like TCP, RDMA Reliable Connections will send ACKs. The ACK will confirm the latest PSN it has received, enabling packet retransmission of missing packets.

|     |              |                |                |       |                            |      |
|-----|--------------|----------------|----------------|-------|----------------------------|------|
| 670 | 11.401488065 | 169.254.114.4  | 169.254.243.96 | RROCE | RC Send Middle QP=0x000081 | 4154 |
| 671 | 11.401602595 | 169.254.114.4  | 169.254.243.96 | RROCE | RC Send Middle QP=0x000081 | 4154 |
| 672 | 11.401605960 | 169.254.114.4  | 169.254.243.96 | RROCE | RC Send Middle QP=0x000081 | 4154 |
| 673 | 11.401615585 | 169.254.114.4  | 169.254.243.96 | RROCE | RC Send Middle QP=0x000081 | 4154 |
| 674 | 11.401627235 | 169.254.243.96 | 169.254.114.4  | RROCE | RC Acknowledge QP=0x000501 | 62   |
| 676 | 11.401755310 | 169.254.114.4  | 169.254.243.96 | RROCE | RC Send Middle QP=0x000081 | 4154 |
| 677 | 11.401758650 | 169.254.114.4  | 169.254.243.96 | RROCE | RC Send Middle QP=0x000081 | 4154 |
| 678 | 11.401768375 | 169.254.114.4  | 169.254.243.96 | RROCE | RC Send Middle QP=0x000081 | 4154 |

# WHAT ABOUT GPU MEMORY?

Currently, the GigE Vision Working Group is focused on integrating RoCE v2 RDMA into the GigE Vision standard. While RoCE v2 enables remote devices to transfer data directly into PC memory, technology exists that allows data to be directly transferred into a GPU's graphical memory. Nvidia's GPUDirect® technology is a high-performance networking technology currently utilized in data-intensive applications such as machine learning, big data analytics, and high-performance computing. The technology enables direct communication between GPUs without requiring intervention from the host CPU. While both RoCE v2 RDMA and GPUDirect bypass the CPU and OS for data transmission, GPUDirect technology distinguishes itself by utilizing the GPU's graphical memory instead of the host PC's memory. This capability allows image data produced by machine vision cameras to be directly transferred to the GPU's graphical memory for accelerated vision processing, leveraging GPU-accelerated platforms and libraries such as Nvidia's CUDA parallel computing platform.



**figure 10: GPU Vision Processing**  
Due to their parallel processing design and numerous cores, GPUs excel in handling image processing tasks that rely on deep learning or machine learning algorithms.

LUCID recognizes the potential of this technology and is collaborating closely with the GigE Vision Standard Working Group to standardize RDMA implementation over GigE Vision, promoting open standards and fostering interoperability. By standardizing these technologies, LUCID and the Working Group aim to ensure that these high-performance networking solutions can be implemented more easily and widely, encouraging innovation, lowering costs, and creating greater accessibility for users across industries.

## UDP VERSUS RDMA BENCHMARKS

As mentioned earlier in this article, one of the primary benefits to using RDMA is to provide multiple 10 GigE cameras streaming with minimal CPU usage. In addition, the data transmissions between camera and host need to be reliable enough to guarantee data packet delivery, similar to TCP. The following benchmark results show that both benefits are achieved. What was tested:

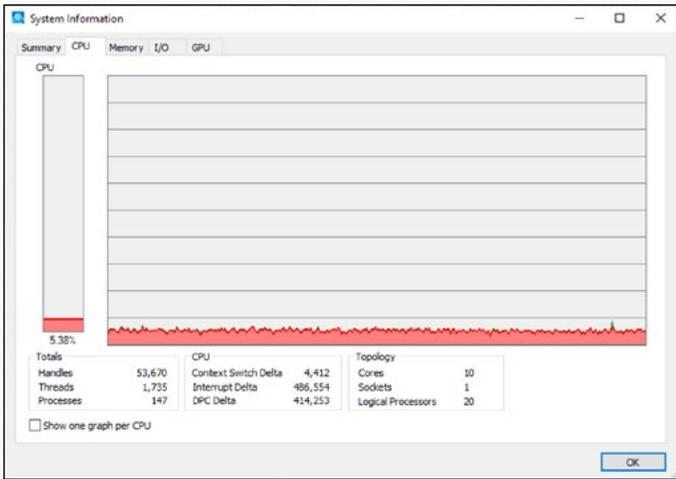
### System Setup

|                                      |  |
|--------------------------------------|--|
| <b>Motherboard</b>                   | ASUS PRIME X299-A II   |
| <b>Processor</b>                     | Intel® Core™ i9-10900X CPU @ 3.70GHz   |
| <b>Memory</b>                        | Kingston 9905743-213.A00G 128GB (8x16GB) 16 GB PC4-25600 DDR4, Quad Channel  |
| <b>Operating System</b>              | Microsoft Windows 10 Professional (x64) version 22H2, build 19045.2604   |
| <b>Video</b>                         | NVIDIA GeForce GT 730, 2GB   |
| <b>Storage</b>                       | KINGSTON SKC3000S512G 512GB  |
| <b>Network Interface<sup>1</sup></b> | 1. LUCID RDMA Host Channel Card (2-Channel 10GigE with PoE+),<br>2. Broadcom BCM57416 (P210tep) Host Channel Card (2-Channel 10GigE), NIC firmware 224.0.159.0 |

### Cameras Used:

|                  |  |
|------------------|--|
| <b>ATX470S-C</b> | Atlas10, 10GigE - 47MP, 8192 x 5556px, Sony IMX492 CMOS, Color   |
| <b>ATX470S-M</b> | Atlas10, 10GigE - 47MP, 8192 x 5556px, Sony IMX492 CMOS, Mono    |
| <b>ATX162S-C</b> | Atlas10, 10GigE - 16.2MP, 5320 x 3032px, Sony IMX532 CMOS, Color |
| <b>ATX162S-M</b> | Atlas10, 10GigE - 16.2MP, 5320 x 3032px, Sony IMX532 CMOS, Mono  |

<sup>1</sup> 2x ATX470S connected to LUCID RDMA HCA (powered using PoE+)  
2x ATX162S connected to Broadcom P210TP (powered externally via GPIO)



**UDP Results:**  
**CPU Usage: 5.38% (avg)**

ATX470S-C: 23.222 FPS (avg)  
 Bandwidth: 8.54576 Gbps  
 Incomplete Frames: 0  
 Host Missed Frames: 0

ATX470S-M: 23.2231 FPS (avg)  
 Bandwidth: 8.54161 Gbps  
 Incomplete Frames: 0  
 Host Missed Frames: 0

ATX162S-C: 68.4799 FPS<sup>2</sup> (avg)  
 Bandwidth: 8.83677 Gbps  
 Incomplete Frames: 0  
 Host Missed Frames: 0

ATX162S-M: 68.5125 FPS<sup>2</sup> (avg)  
 Bandwidth: 8.84099 Gbps  
 Incomplete Frames: 0  
 Host Missed Frames: 0

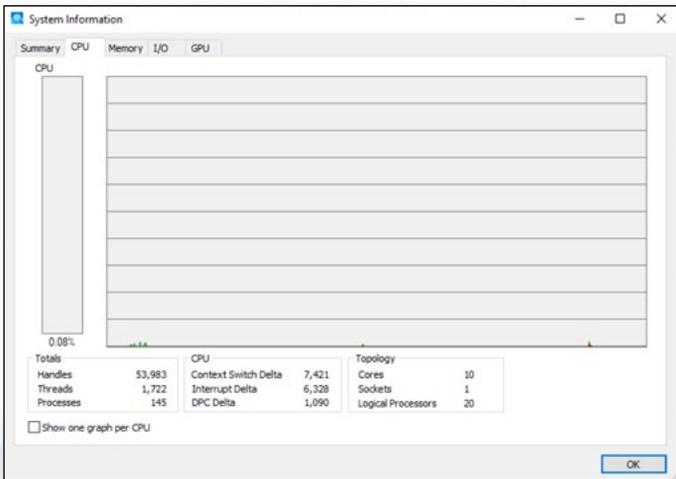
4 Atlas10 cameras were each connected to separate ports on either the LUCID RDMA HCA or Broadcom RDMA HCA. Programming the camera streams were done using Arena SDK. Both LUCID RDMA HCA and Broadcom RDMA HCA firmwares were using version 224.1.102.0. For HCA networking settings in Windows 10, all setting were unchecked except:

- LUCID Vision Labs Lightweight Filter Driver
- Internet Protocol Version 4 (TCP/IPv4)
- Internet Protocol Version 6 (TCP/IPv6)

The ATX470S models bit depth was set to 10 bits and the ATX162S was set to 8 bits.

Cameras were streamed for 24 hours before results were taken.

<sup>2</sup> 10% Device Link Throughput Reserve is set for packet resend, this will reduce maximum FPS for UDP connections. Throughput Reserve is not needed for RDMA connections.



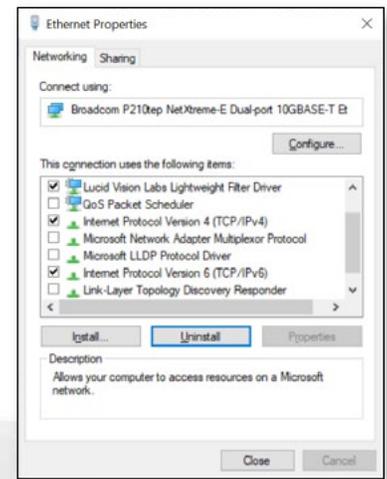
**RDMA Results:**  
**CPU Usage: 0.08% (avg)**

ATX470S-C: 23.232 FPS (avg)  
 Bandwidth: 8.54949 Gbps  
 Incomplete Frames: 0  
 Host Missed Frames: 0

ATX470S-M: 23.2324 FPS (avg)  
 Bandwidth: 8.54935 Gbps  
 Incomplete Frames: 0  
 Host Missed Frames: 0

ATX162S-C: 73.8121 FPS (avg)  
 Bandwidth: 9.52491 Gbps  
 Incomplete Frames: 0  
 Host Missed Frames: 0

ATX162S-M: 73.8442 FPS (avg)  
 Bandwidth: 9.52762 Gbps  
 Incomplete Frames: 0  
 Host Missed Frames: 0



# THE FUTURE IS FAST

Through the utilization of the open standard network technology of RoCE v2, which was initially developed for high-performance computing applications, and adapting it to the GigE Vision standard, LUCID was able to effectively overcome the challenges posed by UDP technology in achieving high bandwidth over 10GigE. As a result, the CPU load generated by image acquisition is decisively reduced, while also providing the lowest possible image delivery latency and the highest data throughput. Ultimately, by incorporating RDMA RoCE v2 into the GigE Vision standard, consumers who opt for Ethernet-based cameras for their high-bandwidth applications will benefit from faster and more dependable data transmission, further reinforcing Ethernet as the preferred industrial interface for machine vision applications.

**10GiGE+RDMA**  
 with Power over Ethernet



#### **LUCID Headquarters**

LUCID Vision Labs, Inc.  
130-13200 Delf Place,  
Richmond B.C.  
Canada, V6V 2A2  
EMAIL: [sales@thinklucid.com](mailto:sales@thinklucid.com)  
PHONE: 1-833-465-8243

#### **Europe, Middle East, Africa**

LUCID Vision Labs GmbH  
Rennalstraße 14, 74360 Ilsfeld  
Germany  
EMAIL: [sales.emea@thinklucid.com](mailto:sales.emea@thinklucid.com)  
PHONE: +49 (0) 7062 97676 12

#### **Asia Pacific**

LUCID Vision Labs G.K  
Eishin Bldg. 4F 3-6-1, Kanda-Ogawamachi,  
Chiyoda-ku, Tokyo 101-0052 Japan  
EMAIL: [sales.apac@thinklucid.com](mailto:sales.apac@thinklucid.com)  
PHONE: +81 3 5577 7915

#### **South Korea**

4F-429, 398 Seocho-daero,  
Seocho-gu, Seoul, South Korea  
EMAIL: [sales.apac@thinklucid.com](mailto:sales.apac@thinklucid.com)

#### **Greater China**

LUCID Vision Labs, China  
51F, Raffles City, No 268 Middle Xizang Road,  
Huangpu District, Shanghai, China.  
EMAIL: [sales.gc@thinklucid.com](mailto:sales.gc@thinklucid.com)

#### **LUCID Vision Labs, Taipei, ROC**

10F., No. 290, Sec 4, Zhongxiao E. Rd., Da'an Dist.,  
Taipei City 106, Taiwan (R.O.C.)  
EMAIL: [sales.gc@thinklucid.com](mailto:sales.gc@thinklucid.com)

**LUCID**  
VISION LABS

© 2023 LUCID Vision Labs, Incorporated. All rights reserved. Helios, Atlas, Phoenix, Triton, Arena, ArenaView and other names and marks appearing on the products herein are either registered trademarks or trademarks of Lucid Vision Labs, Inc. and/or its subsidiaries. v.1.1