PoC Project name:

Reference Implementation Model for the Interactive Live Music Entertainment Use Case

Classification: IOWN Global Forum Recognized PoC

Stage: SSF PoC Report

Confidentiality: Public

Version: 1.0

Feb. 5, 2025

Contents

1. Introduction

In the Interactive Live Music Use Case (ILM UC), the challenges are to reduce the motion-to-photon latency and to improve the efficiency of computational resources allocation for simultaneous connections in the meta-verse space. This PoC for Rendering and Video Delivery (PoC-RVD) [ILM RIM POC REFERENCE] was evaluated specifically for video transmission.

From the motion-to-photon latency perspective, the requirement is to realize the Use Case while detecting Audience's motion information, render it in the cloud, and instantly return the rendered video to the Audience Site. The shorter this time is, the better the user experience will be.

From the viewpoint of simultaneous connections, existing transmission methods use real-time compression to save bandwidth, which increases the processing load of the compression in proportion to the increase in the number of Audiences, resulting in a high server load and high power consumption.

The Interactive Live Music Use Reference Implementation Model (ILM RIM) envisions the use of IOWN Open APN to transmit high-quality video to Audiences with low latency by utilizing wideband transmission. Furthermore, by optimizing the system for IOWN in combination with hardware acceleration such as SmartNIC, the CPU load is expected to be reduced, and the amount allocated to 3D Rendering processing is expected to improve efficiency.

In this PoC report, we will examine the performance of Remote Direct Memory Access (RDMA) over APN which shares GPU memory and Low Latency Media Converter (LLMC) over APN which directly transmits baseband protocols such as DisplayPort/USB over Ethernet. RDMA is expected to take advantage of low latency and high bandwidth, while LLMC is designed to optimize the use of existing video interfaces to achieve low latency. The two systems were evaluated because both architectures transmit raw data in its raw state and are expected to have low latency. It will be compared with WebRTC as one of the existing technologies. We will quantitatively verify what kind of performance is achieved in terms of motion-to-photon latency and the renderer load.

2. PoC Project Completion Status

This PoC focuses on PoC-RVD. Measurements and evaluations are described in section 7.2.

PoC Stage Completion Status: PoC-RVD Significant Step Forward (SSF)

3. Project Participants

PoC Project Name: ILM RIM PoC-RVD Team

- Sony Corporation: Ryohei Takahashi, Toshiya Hamada, Yoshiyuki Kobayashi, Kazumoto Kondo, Hiroshi Kuno
- Fujitsu Limited: Prasad Dhananjaya, Ryoichi Fujinami, Hiroki Takahira
- Sumitomo Electric Industries, Ltd.: Hiroaki Nishimoto, Shinya Uemachi, Daisuke Iiduka, Atsushi Takagi, Tatsuhiko Tanaka
- Keysight Technologies: Kiyoshi Noda, Nobuyuki Kataoka
- NEC Corporation: Tsutomu Tsukagoshi, Kei Mizuno

4. PoC Goals Status Report

- PoC Project Goal #1:
  - Demonstrate if the motion-to-photon latency achieve 10 msec with the Benchmark Model condition by optimizing the data flow with the hardware accelerators. Goal Status (Met)

5. Supported Use Case

Figure 5-1 provides an overview of the PoC-RVD in the ILM RIM. This PoC-RVD focuses on the part enclosed in red line part. This part detects motion information from Audience, renders it in the cloud, and immediately returns the rendered video to the Audience site via Open APN. PoC for Rendering and Video Delivery to Audience Members (PoC-RVD) corresponds to the "motion-to-photon latency" of the ILM RIM Document.

The PoC-RVD is responsible for collecting data from a number of geographically dispersed Audience Member sensor devices, e.g. image sensors, and converting it into the appropriate form, e.g. into Animation data. It also combines it with Avatar Data from the Virtual Space Creator Node, performs Rendering and transmits the rendered video to the Audience Member. The Artist is captured as a volumetric video, which increases the rendering load to generate views for each individual. In the ILM RIM Document, low latency and low power consumption data transfer/processing is realized through the use of high-speed data sharing methods such as RDMA over APN, implemented on Renderer Nodes in Edge DCs.

The aim of this PoC-RVD is to validate the applicability and effectiveness of the solution set for PoC-RVDs and to prove that PoC-RVD with IOWN GF technology can be a building block to realize the ILM UC with low power consumption.

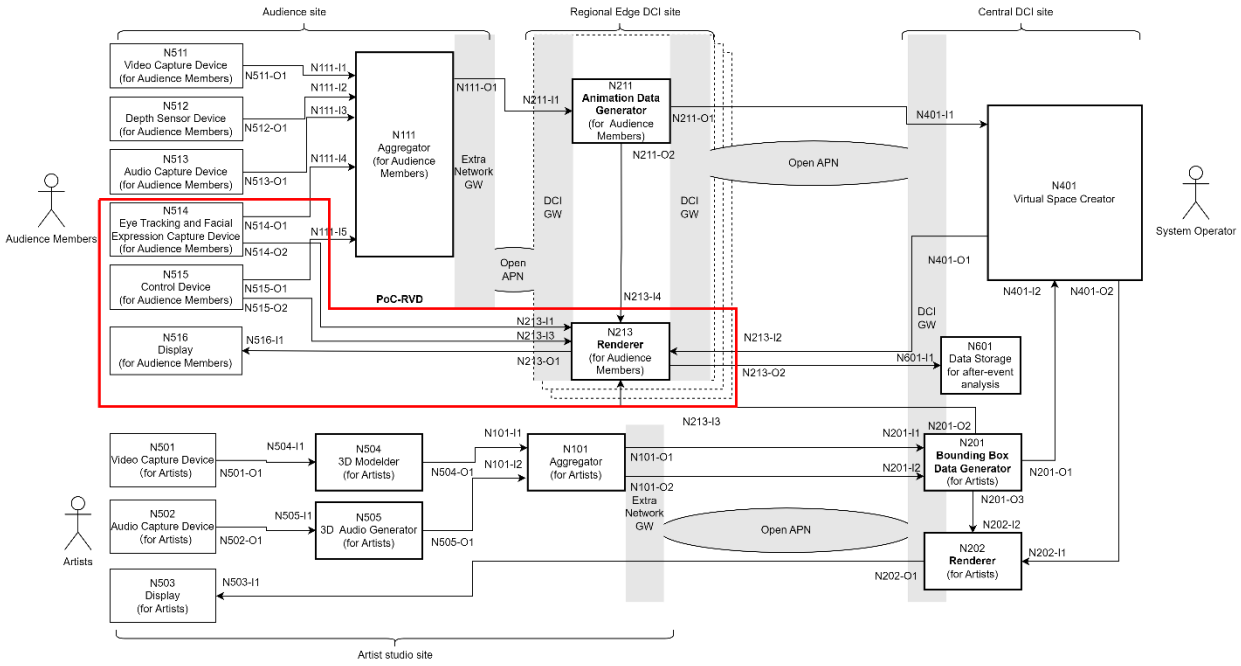

Figure 5-1: Overview of PoC-RVD

6. Confirmation of PoC Demonstration

PoC Demonstration Event Details: March 4-6, 2024, at NEC CONNECT LAB, Abiko city, Chiba, Japan.

Figure 6-2: NEC CONNECT LAB (PoC in Abiko city)

PoC Demonstration Event Details: May 29, 2024, at Sony Osaki, Tokyo, Japan.

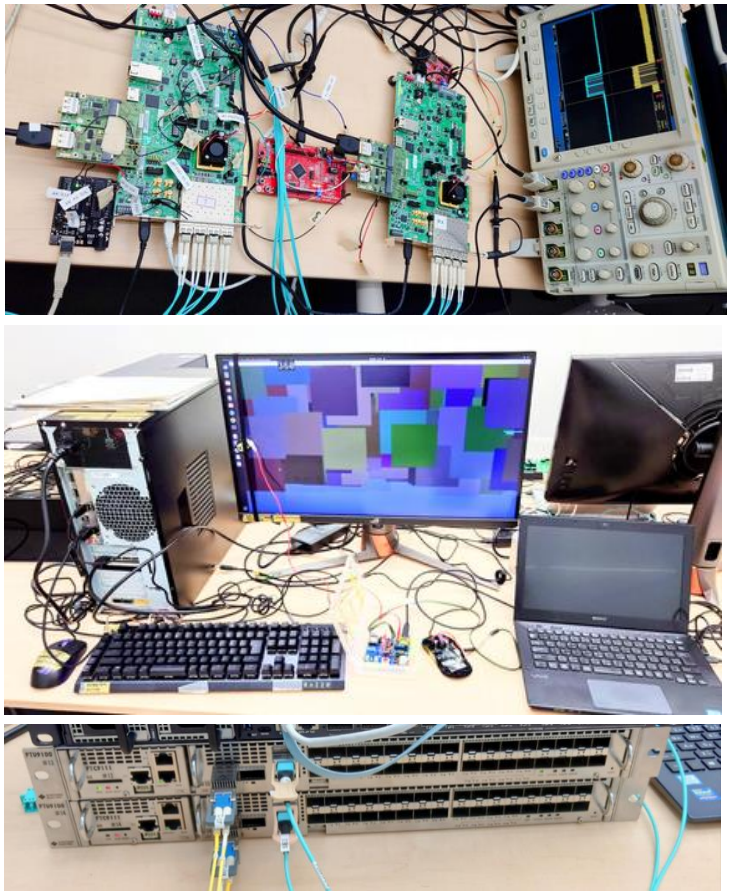

Figure 6-3: Sony-Sumitomo Electric Joint PoC with LLMC (Low Latency Media Converter) and APN Extra-Network Gateway (FTU9100) at Sony City Osaki

7. PoC System Configuration and Implementation

This section describes PoC system configuration. The figure is the basic configuration to evaluate the PoC. See the table in Appendix B for detailed configuration.

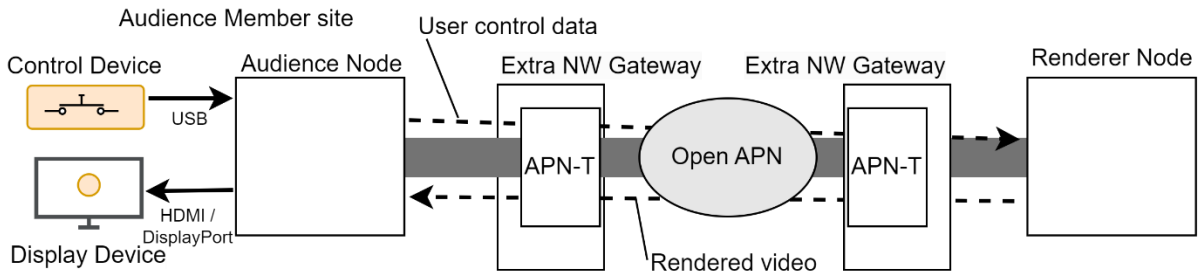

Figure 7-1: Basic configuration

This study aims to assess the effectiveness of motion-to-photon for AV transmission from the Renderer Node to the Audience Site in ILM RIM, when utilizing 100Gbps (HW Accel. by SmartNIC) and the wide bandwidth of IOWN Open APN. The objective is also to estimate the computing resources required to implement the Use Case.

In this PoC report, we will analyze that the developed PoC systems (RDMA and LLMC) can utilize IOWN Open APN to its full potential, while comparing them with legacy WebRTC as a legacy technology. The RDMA system optimizes the data flow and leverages SmartNIC hardware acceleration for uncompressed video transmission. The Media Converter system is a hardware conversion of DisplayPort/USB data to Ethernet and vice versa.

Today's implementation

In the WebRTC system, the image is encoded after rendering the 3D scene on the GPU of the Renderer Node. The data received at the Audience Site is de-packetized, the image is decoded, and the image is displayed via DisplayPort.

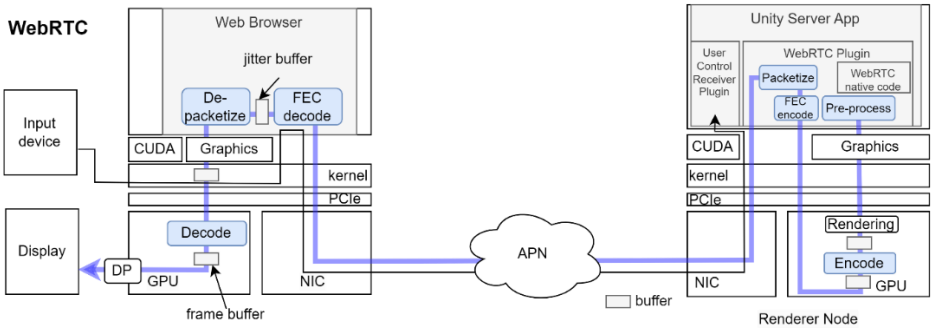

Figure 7-2: WebRTC data flow of AV stream

IOWN based implementation

PoC System (RDMA)

Remote Direct Memory Access (RDMA) is a networking technology that enables direct memory-to-memory communication between computers over a network, bypassing the involvement of the operating systems on either machine. On the other hand, GPUDirect RDMA, developed by NVIDIA, is a technology that enables direct data exchange between a GPU and other PCIe devices like network adapters, storage systems, or even other GPUs, without involving the CPU or system memory.

In this Use Case, we chose to use GPUDirect RDMA because of its strong GPU-dependent processing. We call GPUDirect RDMA as PoC System (RDMA) in this document.

In the PoC System (RDMA), the same process as WebRTC is performed up to the rendering process on the GPU, but then the data is sent to the NIC via the PCI Express bus without going back to the CPU. It bypasses the traditional network stack, allowing applications to directly access the other side's memory. This provides low latency and high throughput for data transfers. The image is then sent to DisplayPort to be displayed on the Display. RDMA was chosen because of its promising low-latency potential.

- The client application detects user input and sends it to the renderer via RDMA
- The rendering node consists of a Renderer application and a plug-in that transfers rendered images via RDMA
- Images are transferred in frame-by-frame bursts, so low latency transmission is expected

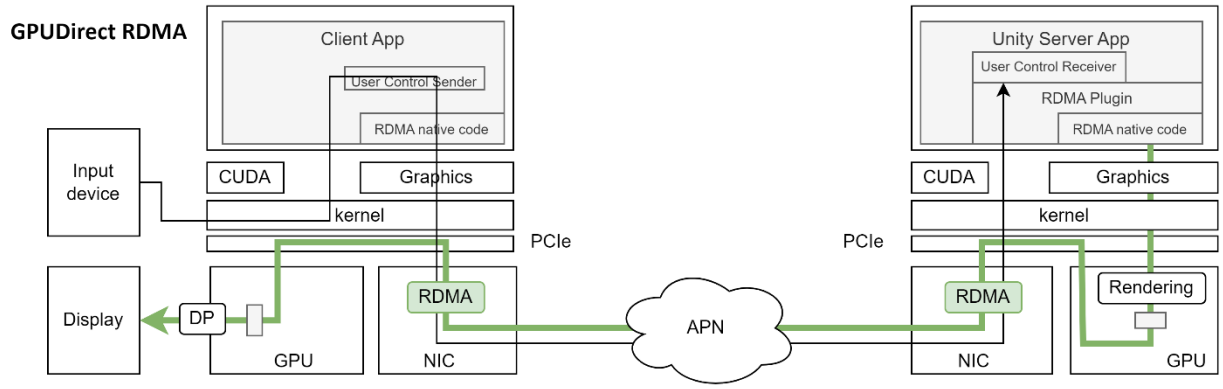

Figure 7-3: PoC System (RDMA) data flow of AV stream

In WebRTC System, as the number of Audiences increases, mainly two elements increase: the first is the Unity rendering process, and the second is the AVC real-time video encoding process. In terms of system load, encoding images generated by rendering does not reduce the frame rate. This indicates that compressing and transmitting images is more efficient in terms of system load.

By compressing the video bit rate to 1/100 or less, the bandwidth required for video transmission can be lowered, i.e., transmission efficiency can be increased. On the other hand, buffering generated by the encoding process on the server side and the decoding process on the client side causes a delay of one frame or more in a typical implementation, which is disadvantageous from a latency perspective. Comparing the transfer rate at 240 fps, the PoC system (RDMA) has a transfer rate of approximately 2 Gbps for uncompressed video, which is 100 times higher than the approximately 20 Mbps of the PoC system (WebRTC). In addition, there is no increase in latency due to memory copying between each process from rendering on the server to display on the client, and buffering during encoding and decoding in PoC System (RDMA). It uses uncompressed video, so there is no degradation in image quality.

PoC System (Low Latency Media Converter, LLMC)

Here, a proof-of-concept system using a LLMC is called a PoC system (LLMC). In the case of PoC System (LLMC), the video signal of the DisplayPort(DP) output from the Renderer Node is converted to the Ethernet packet with UDP/IP protocol by the FPGA of the LLMC (TX) and sent to the downstream Audience Node via Open APN. The Audience Node does not require a PC for display, and the video signal on the Ethernet is converted to the DisplayPort output by the FPGA of the LLMC (RX), and the image is displayed on the display. Meanwhile, the user control data of the USB input from the Input device of the Audience Node is converted to the Ethernet packet with the UDP/IP protocol and sent to the upstream Renderer Node via Open APN. The user control data on the Ethernet is converted to the USB protocol by the LLMC (TX) of the renderer node and passed to the Server Application (Unity) on the CPU of the renderer node. The difference between the LLMC and the media converters currently used for video transmission is low latency and high speed. For this reason, the LLMC in the PoC system (LLMC) does not perform any video compression or decompression, which requires buffering depending on the length of the video frame, and while maintaining the protocol, it takes advantage of the high speed and low latency of Open APN to transparently transmit DisplayPort and USB signals with low latency in micro packets.

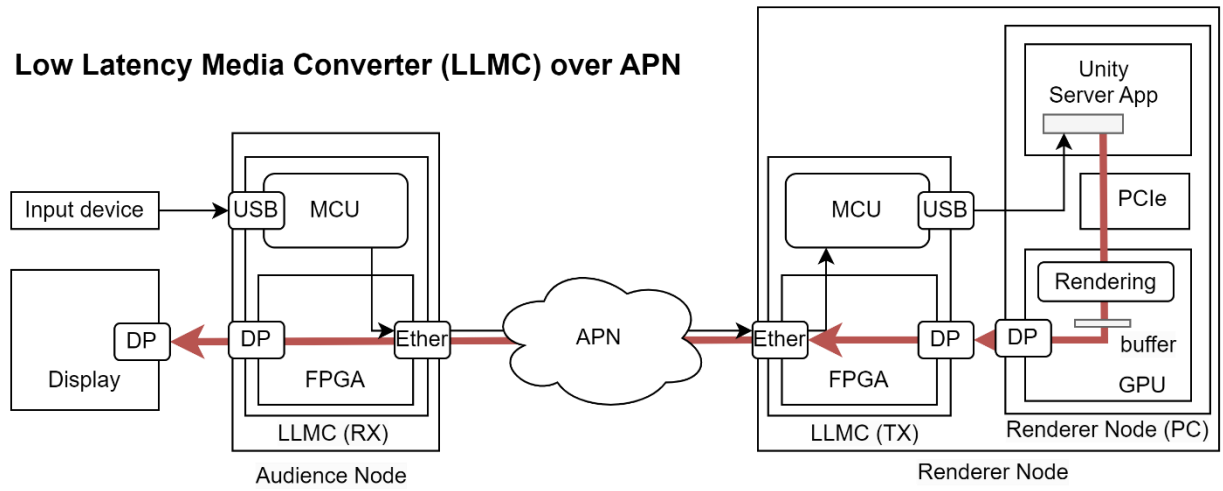

Figure 7-4: PoC System (LLMC) data flow of AV stream and control data

7.1 Overview of implemented PoC System Configuration

This section describes the system configuration used this PoC.

The PoC system (RDMA) and the WebRTC system have almost the same configuration, but differ from the PoC system (LLMC).

PoC System (RDMA) and WebRTC System

The following figure shows the system configuration for the PoC.

- The Renderer Node and Audience Site are connected using NVIDIA's ConnectX-5. Each graphics card is NVIDIA's A4000.
- Keysight Technology's NE3 Network Emulator is connected to emulate the distance.
- NEC's Galileo Flex T is connected to verify the use of Open APN.

- Each of the above network devices is capable of 100 Gbps.

WebRTC has an additional device for using a STUN server. Specifically, a 1 Gbps NIC on each node connects to a router, which in turn connects to an external STUN server.

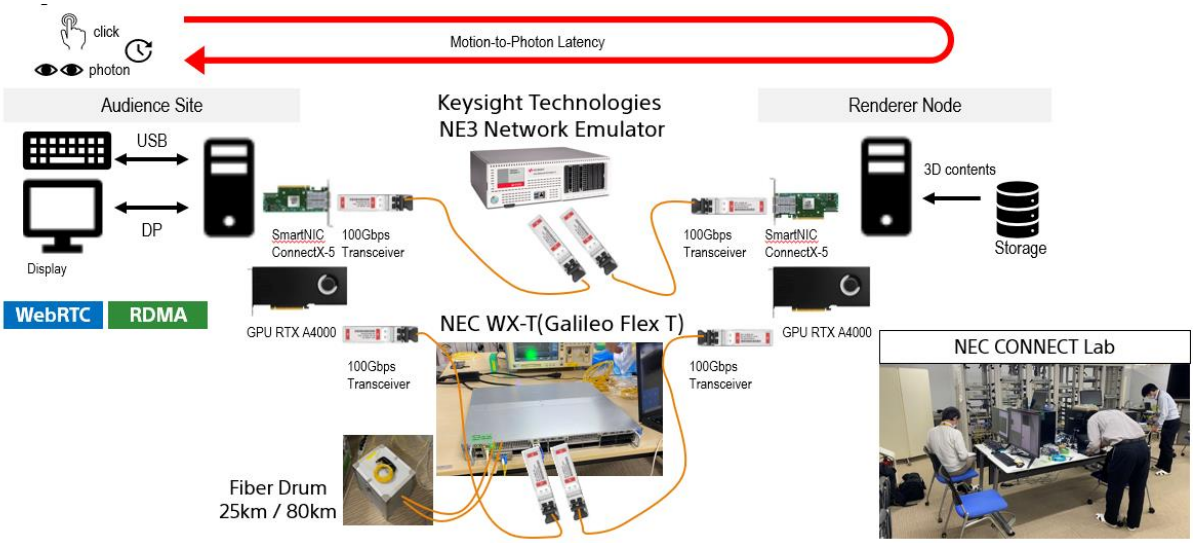

Figure 7.1-1: PoC System (RDMA) and WebRTC System Overview

All experiments with PoC System (RDMA) and WebRTC system in this report were conducted using Keysight's Network Emulator, NE3. As a preliminary check before verification, we used NEC's WX-T to measure the delay of the APN-T device itself. The delay was on the order of tens of microseconds, so we omitted it here.

PoC System (LLMC)

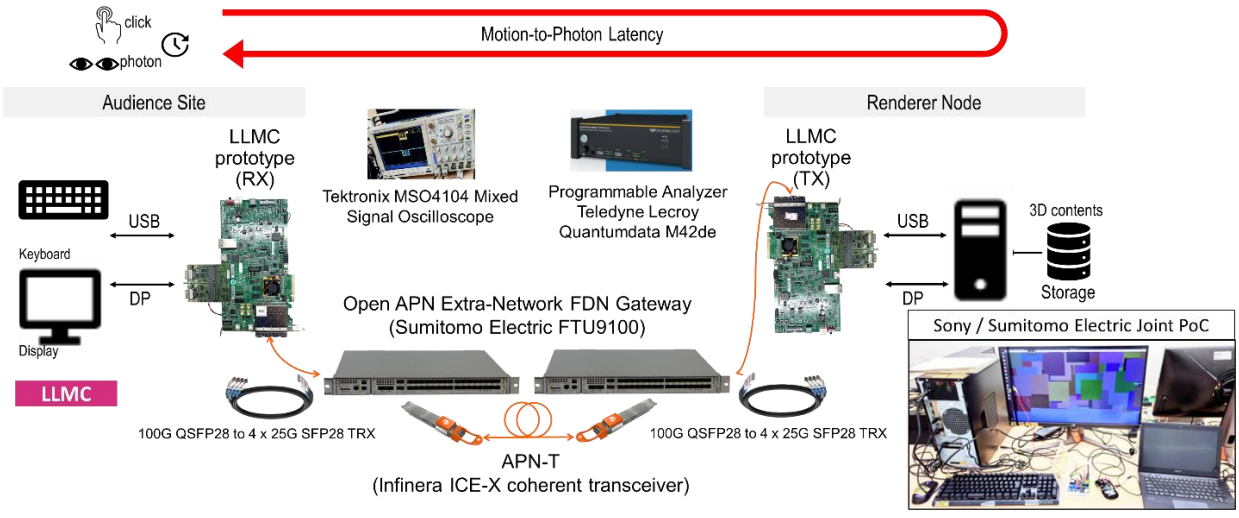

Figure 7.1-2: PoC System (LLMC) Overview

The following figure shows the system configuration and measurement points for PoC system (LLMC).

- A pair of LLMC (TX and RX) prototype, which is described in Table B.3-1 has been newly developed with using the evaluation board of Xilinx XCKU5P FPGA for this PoC System (LLMC).

- Since the ILM is a service which will be provided to multiple of the Audience Site such as venue or home, in the PoC System (LLMC) we used a combination of Open APN Extra-NW Gateway and APN-T which can support not only point-to-point but also point-to-multipoint wavelength path configuration. However in this first PoC system (LLMC) a point-to-point network configuration with 100 Gbps throughput is used.
- For Open APN Extra-NW Gateway, the model FTU9100 Ethernet Switch, which has 400 Gbps switching capacity with 4 x 100 Gbps interface ports and controller function for APN-T, is applied.
- For APN-T, the model ICE-X, Open XR Optics compliant optical transceiver, which supports DWDM and subcarrier multiplexing technology is applied.

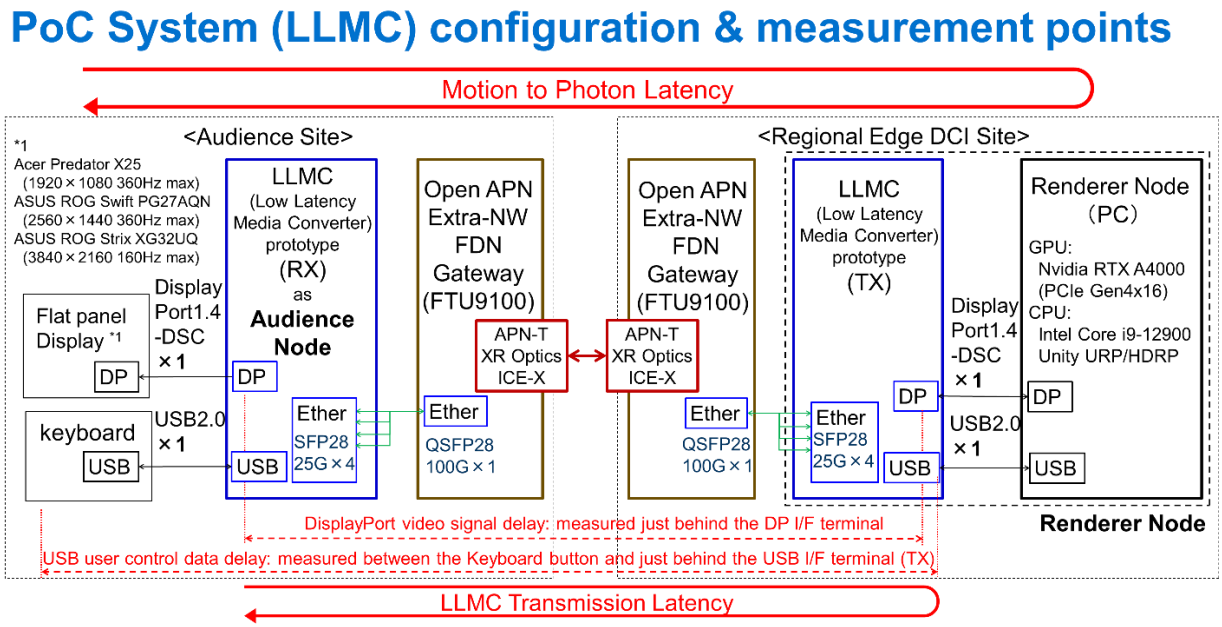

Figure 7.1-3: PoC System (LLMC) configuration and measurement points

- See Appendix B.1 for Renderer Node and Audience Node hardware configuration, software configuration, and application information.
- See Appendix B.2 for Open APN and FDN specifications.
- See Appendix B.3 for LLMC specifications.

7.2 Measurement method

We used a mouse and PC monitor instead of a HMD, because it is not easy to get the motion data from HMD, and the HMD contains a predictive function and cannot measure latency accurately. And, we have created a measuring tool to measure the latency automatically, with an optical sensor, that characteristics is like the human eye.

7.2.1 Motion-to-photon latency

This section describes the methods of measurement used in the PoC.

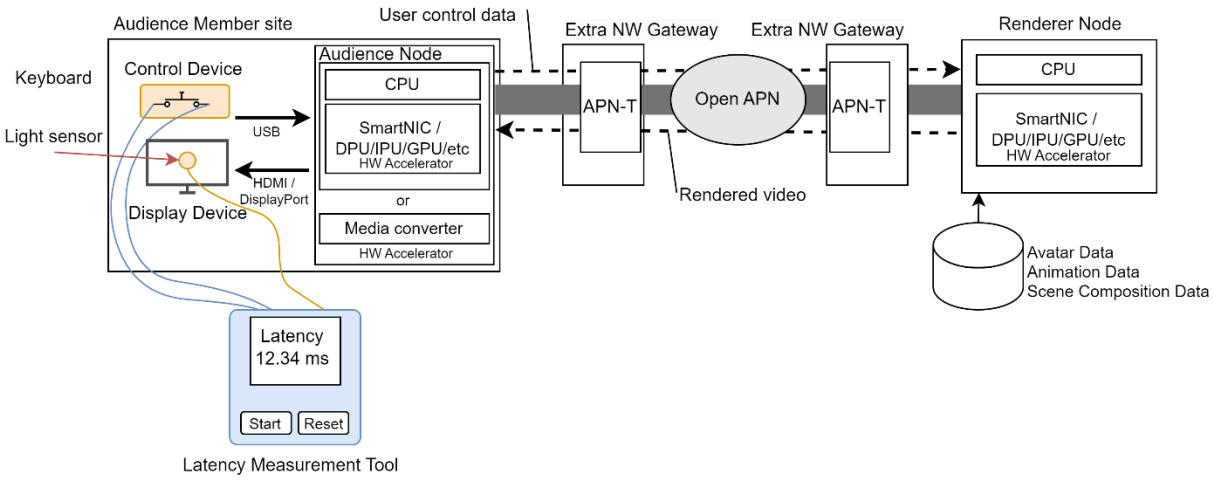

Figure 7.2.1-1: PoC System and Latency Measurement Tools

- Connect the Latency Measurement Tool to your target PC and keyboard to control. The Latency Measurement Tool shorts a key switch on the keyboard as soon as the start switch of the Latency Measurement Tool is pressed, causing the microcontroller to start a timer and terminate the measurement when the brightness exceeds a set threshold.
- Launch an application for the purpose of evaluation that changes the screen when it receives a key input.
- Prepare two images: one with a certain region black, and another with the same region white. Use data values (R,G,B)=(0,0,0) and (R,G,B)=(255,255,255) respectively for black and white color (for 8-bit image). The server changes between the prepared images (from black to white) upon reception of user input.
- The region should be placed in the vertically middle of the screen because the image gradually changes from top to bottom.
- The average luminance level of the measurement region situated in the vertically middle of the display region is measured.
  - The measurement region: The leftmost square in the vertical center of the screen
  - Threshold of the luminance level: 50%. This value is calculated by measuring the brightness of black and white, which is then used as the basis for the 50% value. (White value – Black value) * 0.5 = threshold)

*Note: Take several measurements and find the average as the end result. This is because it depends on the display's timing.

Figure 7.2.1-2: Latency Measurement Tools

7.2.2 Power consumption and CPU/GPU utilization

Table 7.2.2-1: Power Consumption measurement method

Item	Measurement method
CPU	Measure CPU power consumption by the Intel Performance Counter Monitor (Intel PCM) at the same time as motion-to-photon latency measurement $ sudo ./pcm 1 Ubuntu server needs modprob msr. Output the log periodically. Calculate average power consumption after aggregation CPU utilization was taken from the ”C0 core residency” section of Intel PCM. Please see the details in Appendix D.
GPU	Measure GPU power consumption by the nvidia-smi at the same time as motion-to-photon latency measurement $ nvidia-smi -q -d POWER --loop 5 Probe at 5-second intervals. Calculate average power consumption after aggregation. For GPU utilization, the numbers output from the same command were used. Please see the details in Appendix D.
PC total	Read the numbers displayed on Watt Checker, SANWA SUPPLY Watt Checker TAP-TST7

8. Performance Measurement and Evaluation of the PoC System

8.1 Concept to proof

As outlined in the RIM documentation, to achieve a motion-to-photon latency of 10 msec, for example, using a 120 Hz refresh rate display, the time available for rendering is 5.8 msec. However, when taking network latency into account, the available time is further reduced to 3.8 msec, If the network time is assumed to be 1 msec each,

10 msec - Display time (1/120/2 msec) - Network round trip (1 msec x 2) = 3.8 msec

We conducted an investigation to determine if this target latency of 10 msec could be achieved using Open APN.

Table 8.1-1: PoC Objective

Objective Id:	1
Description:	To demonstrate if the motion-to-photon latency achieve 10 msec with the Benchmark Model condition by optimizing the data flow with the hardware accelerators. Goal Status (Met)
Pre-conditions	WebRTC System / PoC System (RDMA) / PoC System (LLMC)

8.2 PoC System (RDMA)

This section describes the result of the PoC System (RDMA). The motion-to-photon latency is 16.08 msec

as you see the Figure 8.2-2. Each value in Figure 8.2-2 was converted to a 4K resolution equivalent to the Benchmark model. Due to the limitations of the PoC system's processing power, the motion-to-photon latency for 3D video rendering was measured to produce a 2K resolution video frame that was then projected to 4K resolution. Please see the details in Appendix C.

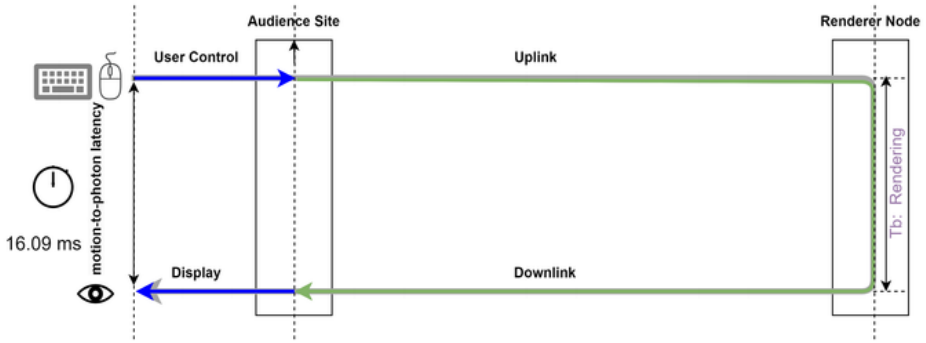

Figure 8.2-1: PoC System (RDMA) motion-to-photon latency

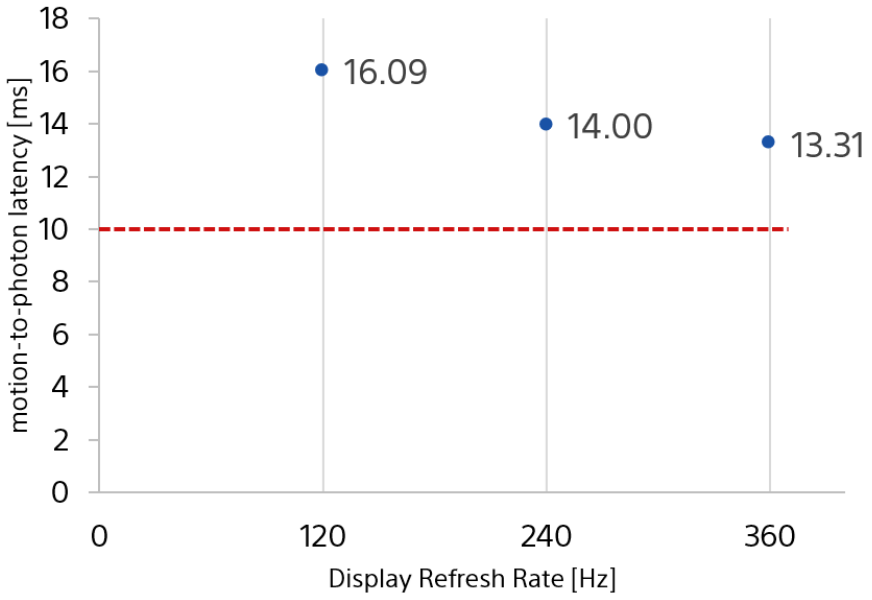

Figure 8.2-2: PoC System (RDMA) motion-to-photon latency

8.2.1 Today's implementation Result, WebRTC System

The findings from testing the existing WebRTC technology are presented. Even though Benchmark Model with a resolution and frame rate of 4K120p, the WebRTC implementation we prepared has upper limits for resolution and frame rate, so we were only able to run it up to 4K60p. With the WebRTC, even the fastest HD resolution of 240p could only reach latency of 75ms, and it is clear that with some improvement, the target of 10 msec cannot be reached. We can also see that frame rate and latency are not directly proportional. HD and 4K refer to resolutions of 1920x1080 and 3840x2160, respectively.

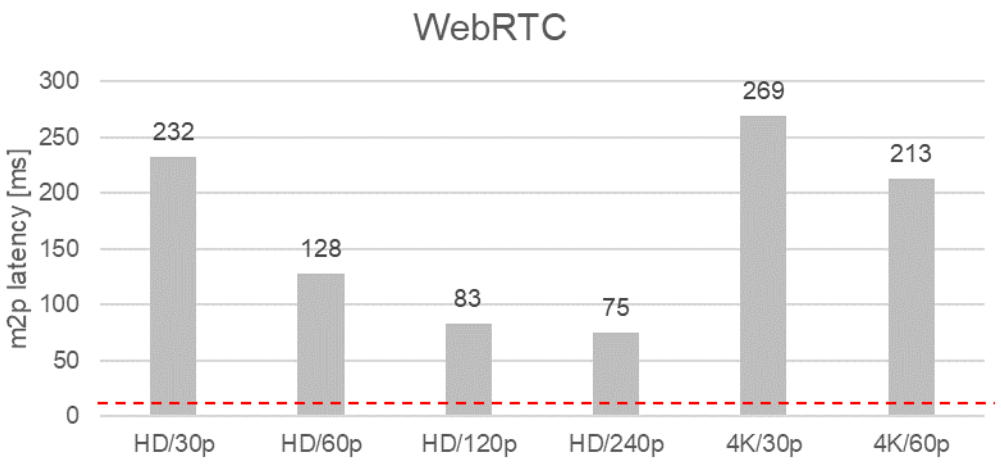

Figure 8.2.1-1: WebRTC motion-to-photon latency

8.2.2 PoC System (RDMA) Latency Breakdown and improvements

As you can see the Figure 8.2.2-1, #1 display part and #2 downlink part takes more than 30% of the total time. Compared to WebRTC System, the PoC system (RDMA) has a transmission bandwidth more than 100 times greater than WebRTC, but shows a marked advantage in terms of transmission speed. For measurement convenience, we split the user control / display process (blue), the transmission and rendering process (green). We divided the processing time by function for ease of measurement, so that we can measure the processing time taken by each segment.

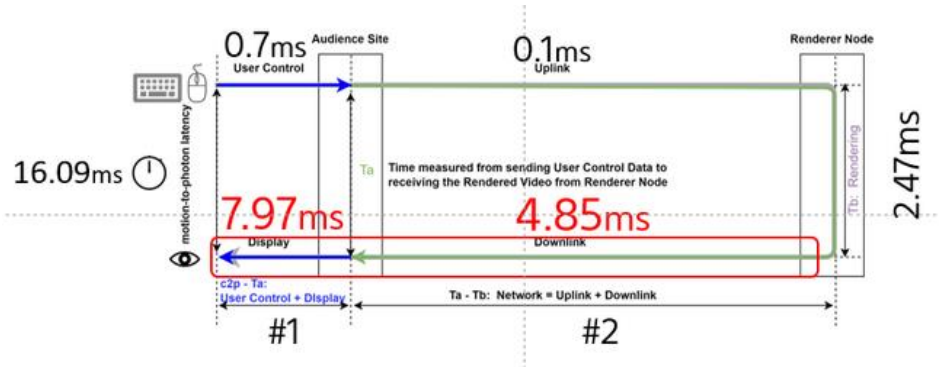

Figure 8.2.2-1: PoC System (RDMA) motion-to-photon latency breakdown

#1. Display Output

With Linux DRM, we found that it was taking a lot of time from CUDA memory to frame buffer as shown below.

The current path is the red line. And this redline takes at least 3.8 msec by our measurement. But it seems this is not optimized, if we change the path in blue line, it is directly to the HW frame buffer, it should be completed in a few 10microsecs. This is the first improvement point.

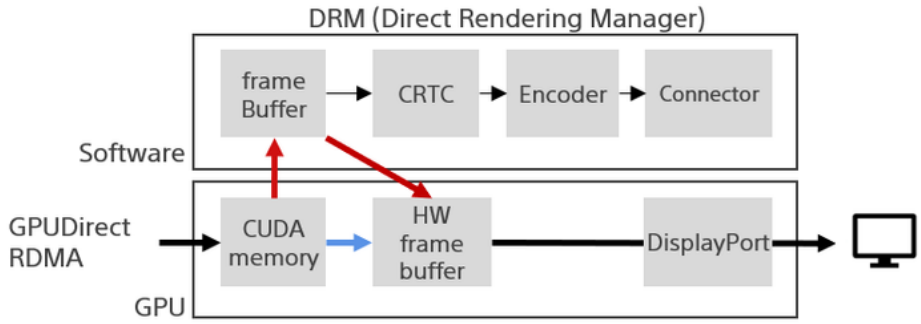

Figure 8.2.2-2: data flow of current implementation and improvement idea

#2. Downlink

As explained in the software stack in Section 7, in this implementation, the generated video image is transmitted from the Renderer Node to the Audience Node in bursts of 4.85 msec per frame. The resolution of the HMD is 4000x2000 per eye, or the equivalent of 4000x4000 for both eyes. The PCI bus width of the hardware used was too narrow to take full advantage of the 100 Gbps network bandwidth. In other words, note that the figures are the result of not fully utilizing 100 Gbps. For example, when applying DSC's one-third compression, which can achieve low latency, the amount of data went down, and the transfer time was reduced to one-third as well, which was estimated to be 1.62 msec, which improves 3.23 msec from uncompressed video. This is the second point to improve. So, we're really excited to see if the DSC in the GPU will be able to output over the network.

Our analysis indicates following approaches can address the bottlenecks to satisfy the latency requirement,

1. Low-latency compression,

2. Optimization to copy directly from CUDA memory (GPU internal memory) to HW frame buffer, instead of copying from CUDA memory to HW frame buffer via DRM frame buffer

If drawing can be done directly from GPUDirect to the frame buffer, unnecessary copying can be eliminated and speed up the process. One idea would be to allow GPUDirect RDMA to output directly to a DRM frame buffer. If the frame buffer is allocated as a CUDA accessible buffer, the copy will be completed in a few 10microsecs.

If these improvements are implemented, latency can be expected to achieve 10 msec,

- Low-latency compression, e.g., DSC, 3.23 msec, this is an estimated reduction time estimated in 8.2.2.
- Optimization to copy directly from CUDA memory (GPU internal memory) to HW frame buffer, instead of copying from CUDA memory to HW frame buffer via DRM frame buffer

Our analysis indicates proposed approaches can address the bottlenecks to satisfy the latency requirement, low-latency video codec and graphics optimization.

For example, at Display Refresh Rate 120 Hz, 16.09 - 3.23 - 3.80 = 9.06 [msec], which means that less than 10 msec could be achieved.

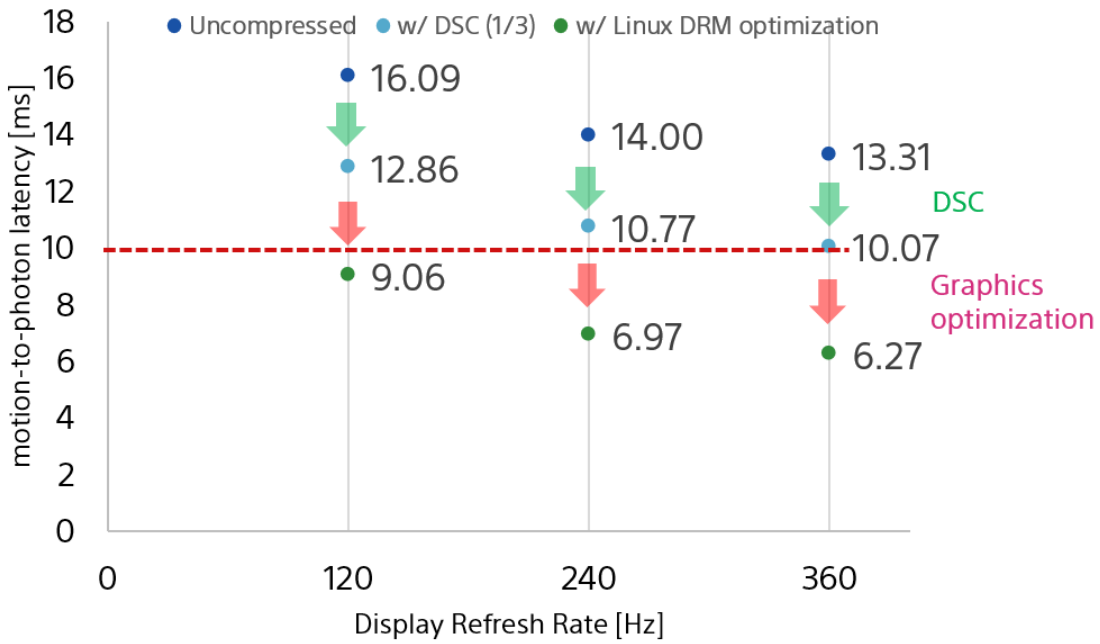

Figure 8.2.3-1: PoC System (RDMA) motion-to-photon latency with improvements

8.2.3 PoC System (RDMA) Distance up to 300km

As you can see the Figure 8.2.4-3, latency increases in proportion to the distance up to about 50 km, and after that, queue related errors, “Send queue full errors” occurs frequently, resulting in a delay in ACK signal from the client. Video frame drops were shown when the distance was over 50 km due to the full of Work Queue in the case of RDMA Reliable Connection. We selected RDMA Reliable Connection because we wanted to assess the capabilities of the standard implementation of RDMA. The relationship between actual distance and error rate is shown in Figure 8.2.3-1.

The reason for this is that the distance increases, the time it takes for the ACK to be returned increases, then the queue fills up and errors occur. For more information on this phenomenon, see Section 2.2 of the Long-Distance RDMA-acceleration Framework, (NTT Technical Review, Vol. 22, No. 3, pp. 75–82, Mar. 2024)

The limit for RDMA video transmission will be done as follows,

Round trip time(Ta) minus Data transmission time should be less than Rendering time

Analyzing the above relationship,

- Theoretical value of rendering time is 1/360 Rendering time, 2.78 msec, Tb in the Figure 8.2.4-2
- The actual measured end-to-end transmission from server to client is 2.40 msec, Ta – Tb in the lower part of Figure 8.2.4-2
- Allowable propagation delay until ACK reception is about 0.38 msec round-trip (0.19 msec one-way), or about 38 km (=190/5, one-way) since the delay is 190 usec. If the PCI bus is capable of 100 Gbps transmission, it can transmit at three times the speed, so it can handle distances of 2.78 - 2.40 / 3 = 1.98 [msec], it would be 1.98 / 2 / 5 = 200 km.
- In the actual measurement, the error was confirmed at 260 usec or more due to the following time settings. In other words, the error was confirmed at 50 km or more. Network emulator was used to adjust the amount of delay based on distance.

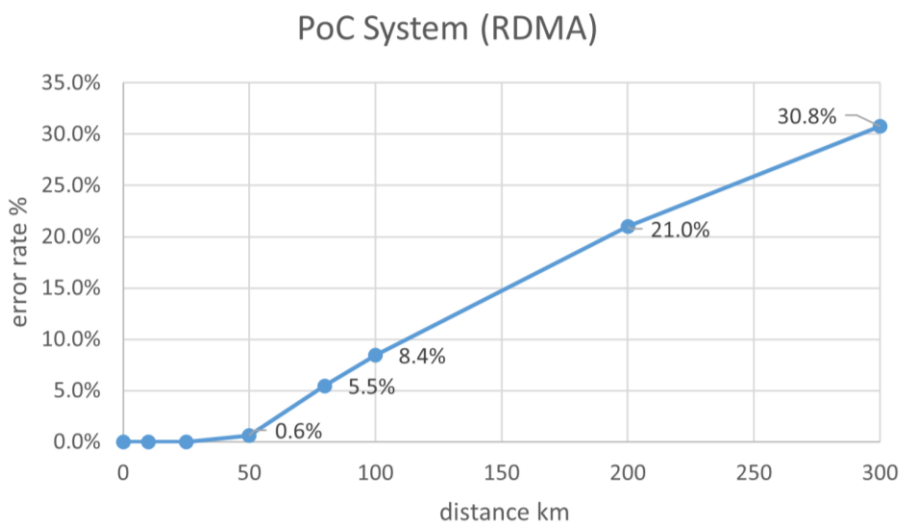

Figure 8.2.3-1: PoC System (RDMA) Distance - Error rate

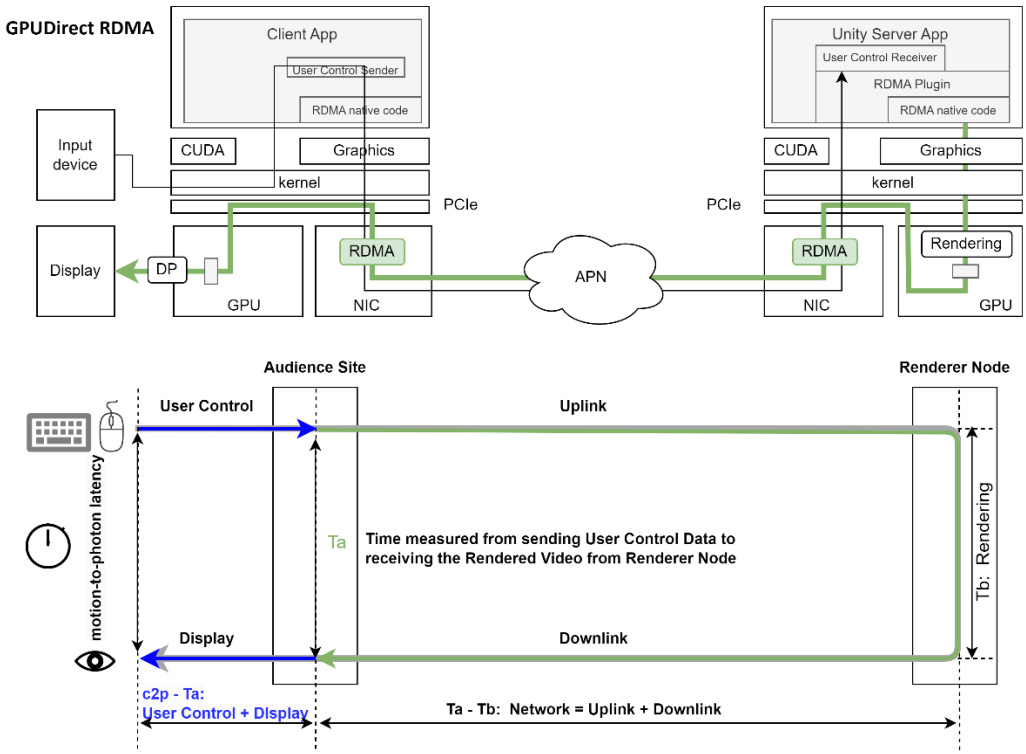

Figure 8.2.3-2: Latency, Definitions of Ta and Tb

Further consideration should be given to the characteristics of the video, such as using RDMA Unreliable Connection instead of Reliable Connection.

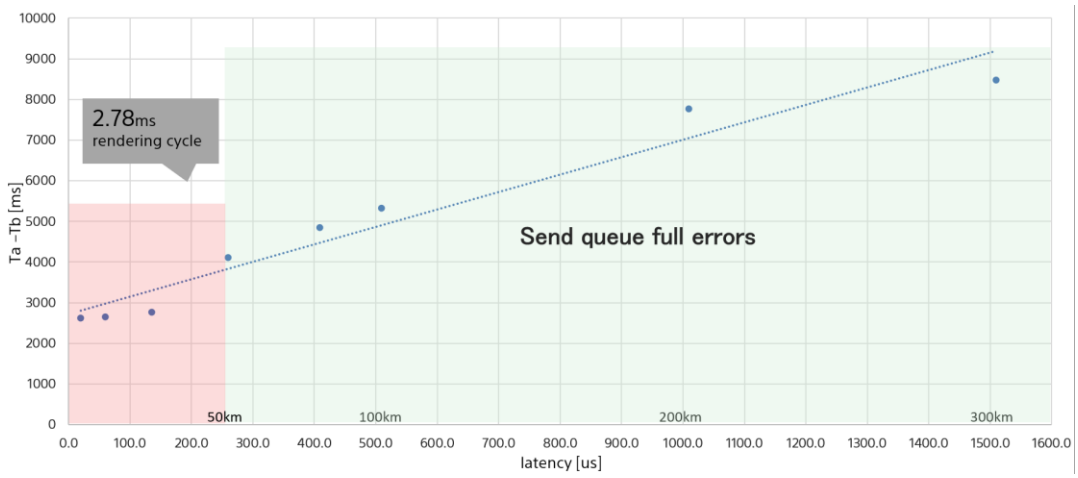

Figure 8.2.3-3: RDMA transmission result

8.3 PoC System (LLMC)

8.3.1 Latency measurement result for PoC System (LLMC)

Here, the results are shown for a Low Latency Media Converter method that converts DisplayPort and USB protocols to Ethernet over APN for transmission and vice versa. (Average value in 100 times measurement)

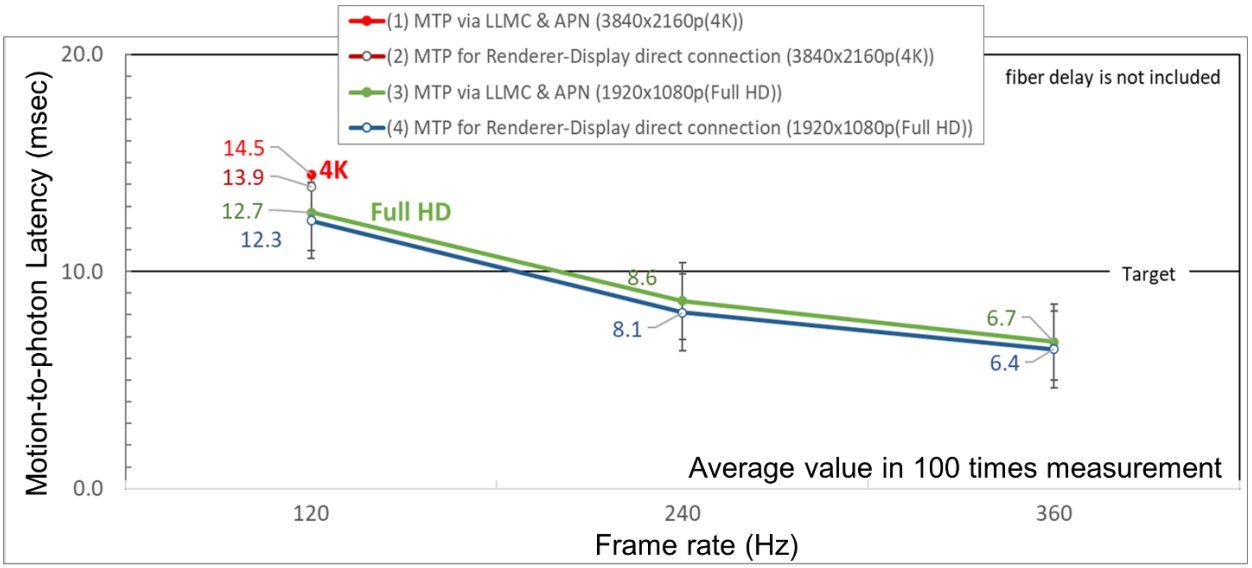

Figure 8.3.1-1: Motion-to-photon latency test result for PoC System (LLMC)

Test results in Figure 8.3.1-1 indicates the followings;

1. A set of motion-to-photon latency via LLMC & Open APN were properly measured for Full HD (1920x1080p) contents as follows;
2. 6.7 msec at 360 Hz,
    1. 8.6 msec at 240 Hz and

2. 12.7 msec at 120 Hz

3. 4K DSC compressed DisplayPort1.4 video was also properly transmitted & displayed with the following motion-to-photon latency;

    1. 14.5 msec at 120 Hz

4. Considering that LLMC uses UDP protocol which is free from ACK delay in long distance, the maximum expected distances from the Edge DC to the Audience which realize the target MTP latency of 10 msec in average are;

    1. 300 km at 360 Hz (∵6.7 msec + (0.5 msec / 100 km) x 2 (round trip) x 300 km = 9.7 msec)

    2. 100 km at 240 Hz (∵8.6 msec + (0.5 msec / 100 km) x 2 (round trip) x 100 km = 9.6 msec)

Note: The effects of measurement variability and manufacturing variability are not taken into account.

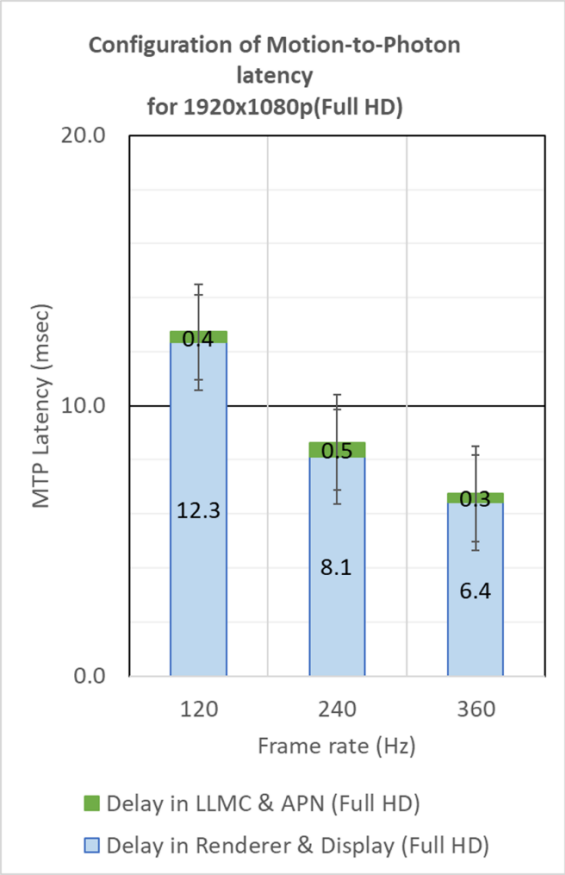

Figure 8.3.1-2: Motion-to-photon latency configuration for LLMC (Full HD)

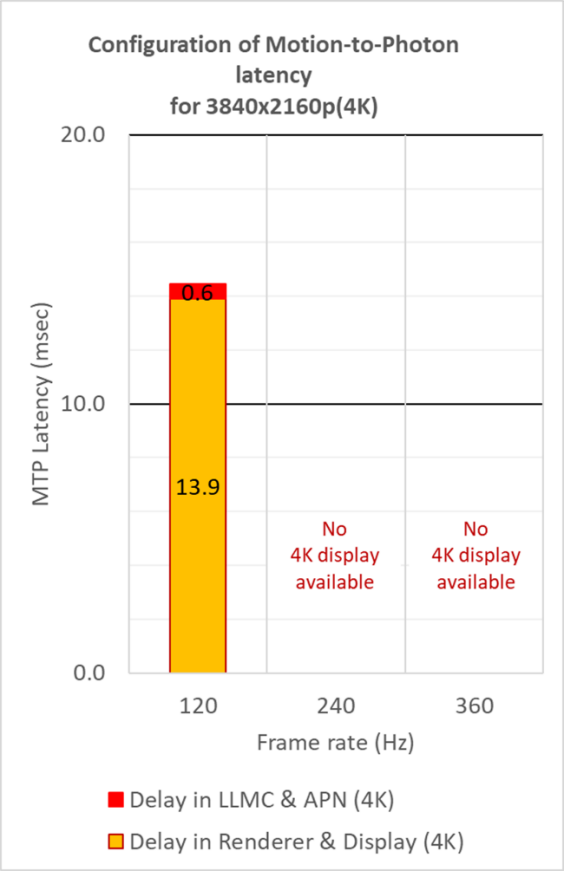

Figure 8.3.1-3: Motion-to-photon latency configuration for LLMC (4K)

To evaluate the configuration of motion-to-photon latency, we divided the motion-to-photon latency into two (2) parts, delay in LLMC & APN and delay in Renderer & Display. The results are shown in Figure 8.3.1-2 and 8.3.1-3. From Figure 8.3.1-2, we can see that;

- In the PoC system (LLMC) 95% to 97% of the motion-to-photon latency occurs in the renderer & display.
- They are highly dependent on the refresh rate of the video stream.
- Judging from the measurement points in Figure 7.1-3 and the data flow in Figure 7-4, it is assumed that the main cause of the refresh rate dependency of the motion-to-photon latency of the Renderer Node and Display is on the buffer management/control of the GPU (card) of the renderer node.
- In the case of the PoC system (LLMC), it was confirmed that the target motion-to-photon delay could be met at high refresh rates of 360Hz and 240Hz. However, if further improvements in motion-to-photon latency are required at low refresh rates such as 120Hz, efforts including improvements to the buffer management/control of the renderer node's GPU (card) are expected to become important.

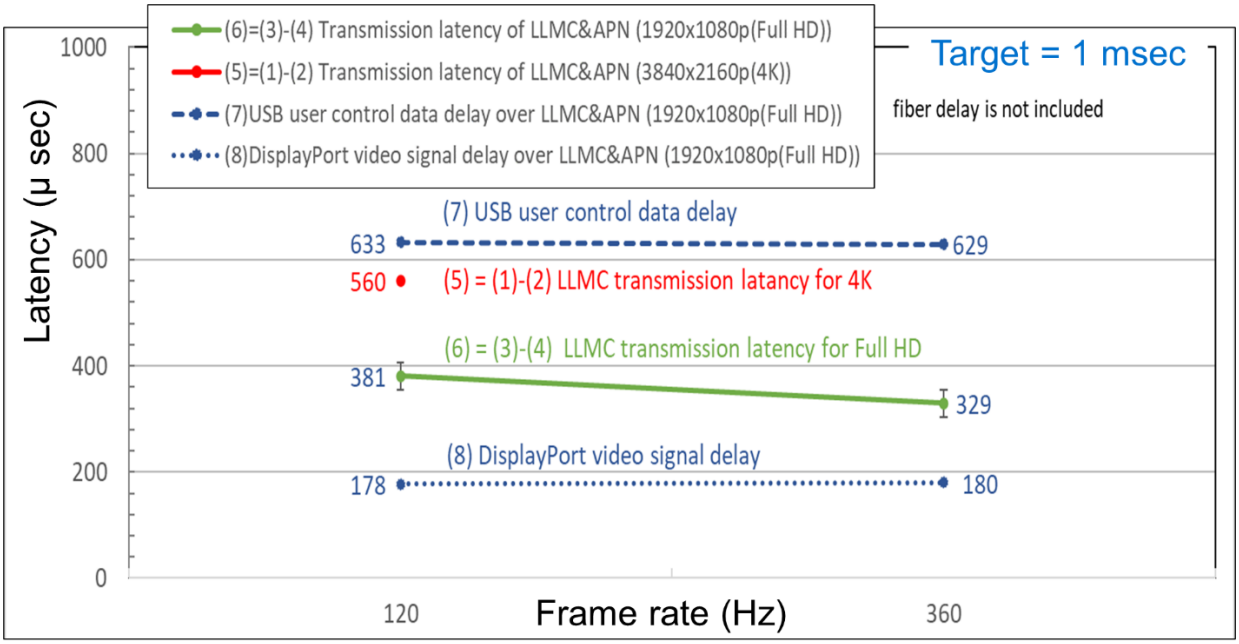

Figure 8.3.1-4: LLMC transmission latency test result

LLMC transmission latency was also measured. The results are shown in Figure 8.3.1-4.

- LLMC transmission latency (round trip delay), which is defined in “Figure 8.1-3 PoC System (LLMC) Measurement configuration”, was calculated from the sum of the measurement result of (7) USB user control data delay (0.63 msec) and (8) DisplayPort video signal delay (0.18 msec). The result was 0.81 msec. It satisfied the target of LLMC transmission latency (round trip delay), 1 msec.
  - Note: LLMC transmission latency does not include the transmission delay of optical fiber (0.5 msec / 100 km for silica core fiber) x 2 (round trip), time lag for Renderer Node and time lag for Display. LLMC transmission latency is a part of “network time” which is defined in section 8.1 that has the target of 2 msec (round trip). The gap between the target for “network time” and LLMC transmission latency is the allowance for the transmission delay of optical fiber from the latency point of view.
- LLMC transmission latency was also reconfirmed by calculating the difference between motion-to-photon via LLMC&APN and motion-to-photon without LLMC&APN. The results were 0.33 msec (at 120 Hz) to 0.38 msec (at 360 Hz) for Full HD, and 0.56 msec for 4K 120 Hz. Judging from the measurement points in Figure 7.1-3, the gap from the 0.81 msec is considered to be the variation of the time lag of Displays.
- It is also confirmed that the LLMC transmission latency is almost free from the refresh rate of the video contents, as it was expected because LLMC does not have any frame size buffer in its TX and RX.

Achievement of POC System (LLMC)

POC System which uses Renderer Node, LLMC prototype and Open APN proved;

- PoC system (LLMC) achieved the 10 msec motion-to-photon latency target at refresh rates of 240Hz and 360Hz, although it did not meet the target at 120Hz.
- Considering that LLMC uses UDP protocol which is free from ACK delay, the maximum expected distances from the Edge DC to the Audience which realize the target MTP latency of 10 msec for Full HD contents in average were estimated as;
  - 300 km at 360 Hz (∵6.7 msec + (0.5 msec / 100 km) x 2 (round trip) x 300 km = 9.7 msec)
  - 100 km at 240 Hz (∵8.6 msec + (0.5 msec / 100 km) x 2 (round trip) x 100 km = 9.6 msec)
- PoC System (LLMC) could properly pass through 4K DSC compressed DisplayPort 1.4 video contents transparently.
- LLMC transmission latency (round trip delay, which does not include the optical fiber delay, time for rendering and display) also satisfied its target of 1 msec. It is almost independent from the video frame rate.
- Note: These results were measured using the same hardware as the other PoCs, but for implementation reasons, the measurements were taken on a Windows OS for rendering server, so a direct comparison with the test results of RDMA which were taken on a Linux OS is not meaningful.

8.4 Consideration

WebRTC, RDMA, and LLMC each have their advantages and disadvantages, and it is not productive to compare simply motion-to-photon latency.

WebRTC has a large motion-to-photon latency, which is far from the target value, but the efficiency of compressed transmission is high and effective in the current situation where transmission capacity is not sufficient.

Although RMDA requires a GPU and SmartNIC for the client, it is able to achieve the motion-to-photon latency target value, and can be said to be able to utilize lines with large transmission capacity, such as Open APN.

LLMC has the advantage of being very easy to use and already achieved the motion-to-photon latency target value at high refresh rate, however considering the implementation on the server side, the current implementation of LLMC requires the server to provide DisplayPort outputs for the number of clients.

8.5 Scalability consideration for Large Scale Audiences

Now that we have verified the motion-to-photon latency with a peer-to-peer connection, we will look at the performance with multiple connections. The Benchmark model's 4000 x 2000 resolution at 120 fps does not meet the desired motion-to-photon latency with the PoC System described in Section 7.1, 10 msec, because the actual rendering speed decreases as the number of clients increases. Prioritize increasing the number of clients while maintaining performance, we will use the results of 60fps/30fps, where the performance of the set target fps is properly achieved.

The way to handle multiple Audiences was to implement multiple cameras in a single Unity process in order to increase the number of clients, rather than running on multiple processes or instances. This was a more efficient solution than running on multiple processes or instances.

From here, the results from PoC System (RDMA) are shown in comparison to WebRTC System.

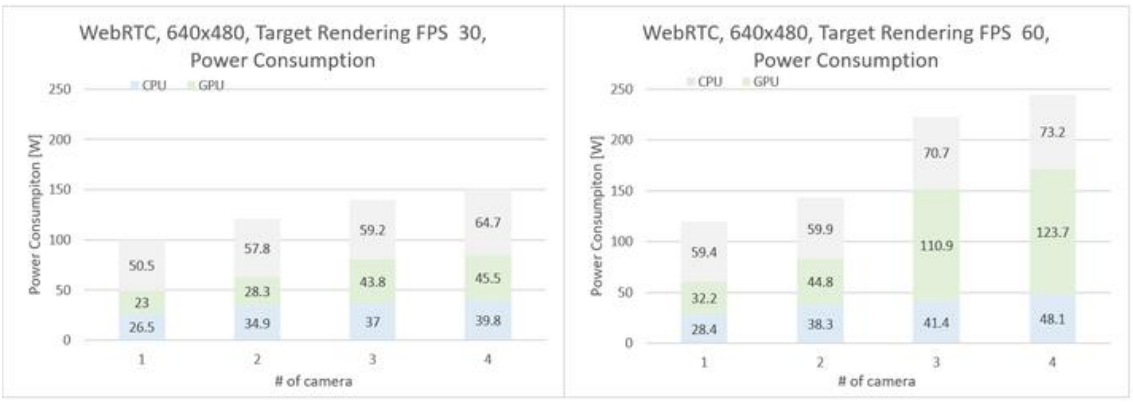

Figure 8.5-1: WebRTC Renderer Power Consumption, 60/30fps

Both 30 fps and 60 fps rendering performance are achieved as per the targeted values. However, because the power consumption at 60 fps was rising discretely with the increase in the number of Audiences, we also measured 30fps, which consumes less power.

Note: Due to implementation limitations, the hardware is the same, but WebRTC runs on Windows and PoC System (RDMA) runs on Linux. Since the operating systems are different, a strict comparison cannot be made, but a trend can be seen.

At 30 fps, PoC System (RDMA) increases GPU power consumption monotonically. It seems that the higher CPU utilization in PoC System (RDMA) compared to WebRTC is probably due to the polling process. More specifically, since the verification was done with the highest priority on achieving low motion-to-photon latency, we implemented a busy-polling mechanism for four clients regardless of whether the client is connected or not, and there is a thread to get the Render Texture in Unity and a thread for RDMA WRITE, which controls the processing order in the pipeline. When the number of client connections is 0, a busy loop occurs in RDMA WRITE to wait for RDMA WRITE execution by polling.

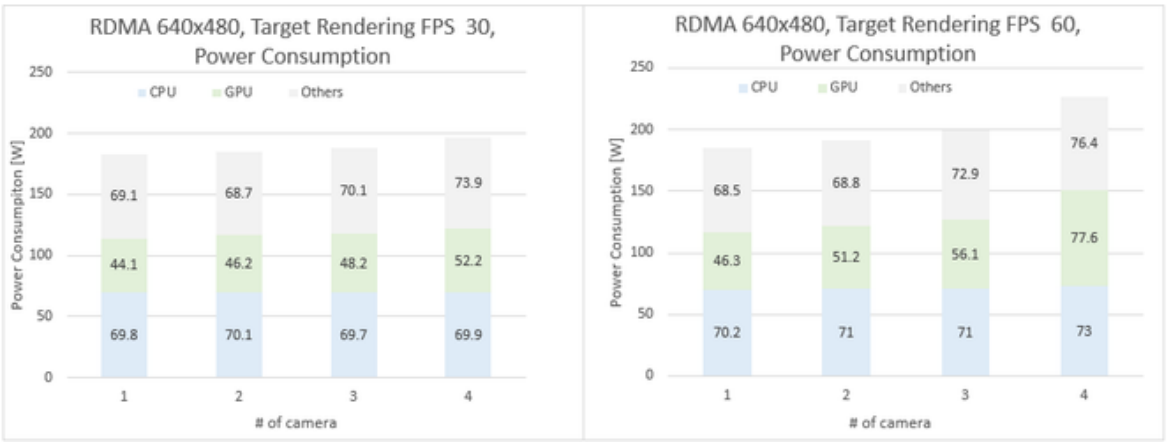

Figure 8.5-2: PoC System (RDMA) Renderer Power Consumption

Although PoC System (RDMA) can achieve 100 times the transmission data rate of WebRTC, the increase in CPU power consumption of PoC System (RDMA) is RMDA 70 / WebRTC 40 = 1.75 compared to WebRTC and client number 4, an increase of only about 75 %.

The comparison focuses on changes in GPU power consumption,

In WebRTC at 30 fps, extrapolating the GPU power consumption for one more person is

- (45.5-23) / 3 = 7.5 W per person

- 8 W (23-15) for the very first 1 person (idle: 15 W)

Since the maximum power consumption of the GPU is 135 W

- (max 135 – 23) / 7.5 + 1 = 15.9, This means it can support up to 15 Audiences

Similarly, in PoC System (RDMA) at 30 fps,

- (52.2 – 44.1) / 3 = 2.7 W per person
- (max 135 – 44.1) / 2.7 + 1 = 34.7, This means it can support up to 34 Audiences

At the same power consumption, for example 135 W, which is the maximum power consumption of the graphics board used for this PoC, the number of Audiences that the POC System (RDMA) can handle is 34/15x100 = 227 %, which is more than twice that of the WebRTC System. In other words, the average power consumption per person in PoC System (RDMA) is less than half of WebRTC System. In the comparison of the rate of increase in power consumption, WebRTC is affected by encoding, and as the number of clients increases, the GPU load tends to increase. Looking at the WebRTC 60 fps graph, we can see that the increase in fps from 32.2 to 123.7 is a clear indication of this effect.

IOWN Open APN can transmit uncompressed data using PoC System (RDMA), taking advantage of the high bandwidth, so the load on the GPU is reduced without encoding, and as a result, it is easy to increase the number of people connected to the system.

Note: Power consumption may be reduced if PoC System (RDMA) User Control Data is not transmitted over RDMA, or if video is optimized to be transmitted over RDMA Unreliable Connection due to omitting the polling process.

Next, we have taken a deep dive into the constant CPU power consumption of PoC System (RDMA) in order to consider the scalability. The following charts show the CPU utilization and GPU utilization of the PoC System (RDMA) Renderer ode.

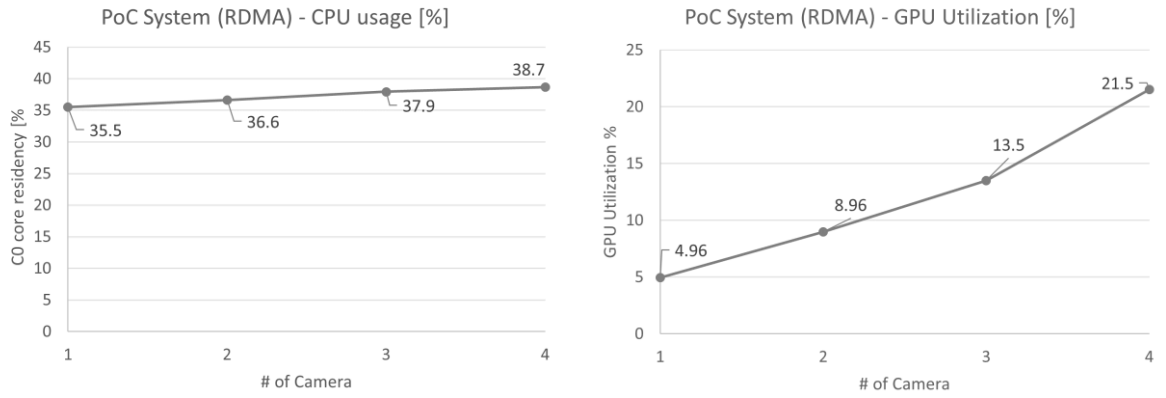

Figure 8.5-3: PoC System RDMA CPU usage and GPU Utilization

We use the number called “C0 core residency” of Intel PCM tool as a CPU usage [%]. See the details in Appendix D. As the graph shows, CPU usage increases only slightly as the number of people increases. This characteristic is helpful for scalability as it indicates that resource consumption increases only slightly when the number of Audiences increases.

As shown in the PoC System (RDMA) graph, the GPU utilization rate shows an increase in proportion to the number of people, while GPU power consumption is slightly higher and not much different. This may be due to the fact that GPUs have already changed to the higher Performance Level. The power consumption of a GPU depends on both Performance Level and utilization, so it must be looked at in a comprehensive manner. Depending on the objective, the analysis should look at both GPU power consumption or GPU utilization.

Since PoC System (RDMA) CPU utilization does not change when the number of connected clients is increased from one to four, it is not affected by the number of clients and can be said to scale easily. On the other hand, GPU utilization increases as the number of clients increases, and the GPU fills up before the CPU does, so the number of clients that can be handled is estimated by extrapolating the GPU utilization figure.

- GPU Utilization [%] = 5.416 * # of Audiences - 1.31
- # of Audience = 19.7, 19 Audiences in integer

The Benchmark Model has a resolution of 4000x2000 with a refresh rate of 120 Hz, while the current verification is at 640x480, 30fps, so there is a gap of about 104 times the data rate. Therefore, the required GPU performance is 5.5 times for transmission, 104 divided by 19, assuming a linear change.

As already explained, the 3D model size of the Benchmark Model itself cannot be processed with the current PC performance, so the verification was conducted with reduced resolution and FPS in order to process the current available hardware.

However, in the case of implementation as per Benchmark Model, as the resolution and FPS increase, one NIC will not be able to handle the bandwidth, so multiple NICs will be required. Furthermore, due to the current Unity implementation limitation of sending data in order, there is a time delay, which affects the motion-to-photon latency. In other words, the more clients that are sent in sequence, the worse the motion-to-photon latency.

There are also various rendering techniques to reduce GPU utilization, including Foveated Rendering for HMDs and LOD techniques for 3D CG. As verified in the RIM document, the LOD technique has proven to be a very effective for this Use Case.

Performance issues are expected to be resolved through the implementation improvements described above and future improvements in computing performance.

9. PoC’s Contribution to IOWN GF

Contribution	WG/TF	Study Item (SI) / Work Item (WI)	Comments
	RIM TF		This PoC demonstrate IOWN GF-based video transmission systems that process 3D rendering in response to user interaction.

10. PoC Suggested Action Items

There are no specific items.

10.1 Gaps identified in relevant standardization

There is no particular connection to standardization.

11. Conclusion

In this PoC, two IOWN GF-based PoCs system (RDMA, LLMC) for video delivery were evaluated by fully utilizing Open APN and HW Accelerator to meet the motion-to-photon latency requirement while maintaining high quality video content.

PoC System (RDMA) did not achieve the target latency with the hardware used for PoC, but can achieve the target with the proposed improvements.

Also, feasibility for motion-to-photon latency requirement, distance up to 300km under the condition defined in PoC Reference document are conducted. The video frame drops were shown over 50km due to the full of Work Queue as expected by the mechanism. However, it is possible to use the edge data centers in each region and transmit using an RDMA Unreliable Connection that does not depend on the distance.

From a scalability perspective, PoC System (RDMA) is more efficient than WebRTC, and can support Audiences with less than half the power consumption of existing technologies in terms of power consumption. It was also confirmed that CPU usage of PoC System (RDMA) stays almost constant and is easy to scale.

PoC System (LLMC) has the advantage of transmission latency (excluding rendering time, display delay and fiber transmission delay) of less than 1 msec, and the ability to support long distances.

PoC System (LLMC) achieved the motion-to-photon latency goal by increasing the refresh rate of the display. On the other hand, instead of providing DisplayPort outputs and placing the LLMC hardware on the server side, the LLMC functionality could be flexibly implemented on the server side so that DisplayPort outputs can be routed to the network via SmartNICs in order to scale the ILM service.

Additional verification will be required to determine if further efficiency gains can be realized through distributed processing in the future.

References

[ILM RIM]	Reference Implementation Model (RIM) for the Interactive Live Music Entertainment Use Case https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-TS-RIMforInteractiveLiveMusicUseCase-1.0.pdf
[ILM RIM POC REFERENCE]	PoC Reference: Reference Implementation Model for the Interactive Live Music Entertainment Use Case https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-RIM_for_ILM_UC_PoC_Reference-1.0.pdf
[IOWN Open APN]	IOWN Global Forum, "Open All-Photonic Network Functional Architecture," 2022.

Abbreviations

0-9
3D, Three Dimensional
B
bps, bits per second
D
DMA-BUF interface, Direct Memory Access Buffer interface
DC, Data Center
DCI, Data-Centric Infrastructure
DPD, Data Pipeline Diagram
DSC, Display Stream Compression
F
fps, frames per second
H
HMD, Head-Mounted Display
I
ILM UC, Interactive Live Music Use Case
ILM RIM, Reference Implementation Model of the Interactive Live Music Use Case
IPU, Infrastructure Processing Unit
L
LLMC, Low Latency Media Converter
R
RDMA, Remote Direct Memory Access
RTP, Real-time Transport Protocol
RVD, Rendering and Video Delivery
S
SDI, Serial Digital Interface
ST2110, SMPTE ST 2110 Professional Media Over Managed IP Networks
T
TCP, Transmission Control Protocol
V
VM, Virtual Machine

Terms and Definitions

Virtual Space	Virtual Space is a virtual live music venue. Audience Members can move freely around the live music venue, and the images seen by each participant are individually generated by rendering from each viewpoint. This is conceptual, and the data is represented by separately defined Scene Composition data and visualized by the rendering process in the renderer.
Audience Member	Person who participates from their homes or karaoke rooms.
Scene Composition Data	Output data from the Virtual Space Creator. It is some part of the elements necessary to construct Virtual Space, which is visualized by the Renderer. The elements of Scene Composition data are as follows; Artists' Bounding Box data (includes determined position and direction) Audience Members' Animation data (includes determined position and direction) Concert Hall component such as 3D model of stage, seats, light position, speaker position, etc. 3D audio data for Virtual Space Computer Graphics effects including lighting and ROI (Region of Interest) Information and provide recommended view port for Audience Member They are expressed in a scene description language and contains both static and time-varying dynamic information. In the 3D scene generation phase, the 3D models are not necessarily required, but only the position and motion information (Bounding Box data and Animation data: time-varying dynamic information) of the 3D models can be used to construct the 3D scene. In the DPD, multiple inputs arrive at the Renderer. Some of the static information can be sent to the Renderer in advance, before going live. They are then integrated before being used for rendering.
SmartNIC	A programmable accelerator that makes data center networking, security and storage efficient and flexible.
Avatar Data	3D model data
Animation Data	Time series data of position and direction of joints to move the Avatar data. The Animation data will be sent to the system in real-time during the live event to move the 3D model, Avatar data. This Animation data includes not only body movements but also facial expressions and eye movements.
Low Latency Media Converter	Low Latency Media Converter is a type of protocol and media converter device which converts dedicated interface such as DisplayPort (or HDMI) and USB to Ethernet(/IP/UDP/RTP) and vice versa. It can be PC based product or dedicated hardware device, however the low latency feature is essential, because non-negligible amount of the Motion-to-Photon or Click-to-Photon latency can be occurred in the Audience Node(s). It can be a device with or without codec function, however the codec should be open standard based.

Appendix.A Requirements and Expectations in the PoC Reference

The table B-1 shows the requirements and expectations in the PoC Reference related to our PoC scope, and how they were treated in this PoC Report.

Table B-1

#	Requirements and Expectations	PoC Report
1	[2.1.2.1.2] Renderer Node Describe data format for Animation Data, etc. in detail Describe the configuration on Virtual Space in detail (The number of Audience Members etc.)	Described in Appendix B (Due to the use of content that lightens the rendering process)
2	[2.1.2.3.2] Scalability Estimate how many Audience Members can be supported by one Render Node with a pair of CPU and GPU. Estimate how many times higher performance Render Node with a pair of CPU and GPU are needed to support a certain number of Audience Members	Described in 8.5
3	[2.1.4.1] Latency Measure Click-to-photon or motion-to-photon latency [0.1 msec order] Latency at different communication distances (0, 10, 25, 50, 80, 100, 200, 300 km) Test results with Network Emulator and Open APN Describe Breakdown of the components comprising the latency, e.g. rendering, network, display and input delay. The display and input delay may be estimated from the device specification. Analyze each latency component in detail and explain possible mitigation technology to meet the requirement. Describe measuring method in detail.	Described in 7 Described measurement method in 7.2
4	[2.1.4.2] Required System Resources and Configuration Describe the environment used for the evaluation (Network and system configuration, including devices, servers, and networks) Describe the system components, such as the number of devices, number of CPUs, memory size, and bandwidth.	Described in 7 Described measurement method in 7.2
5	[2.1.4.3] Energy Efficiency The total energy consumption of the system resources Measure the power consumption of Renderer Node (Wh), Audience Node (Wh), and Audience Member (Wh). Measure the power consumption of Network equipment (Wh), (optional) Describe measuring method in detail.	Described in 8.5 Described measurement method in 7.2
6	[2.1.4.4] Network Resource The network bandwidth resource allocation and consumption, QoS and latency in Access Network between edge DC and physical venue for the PoC Describe the bitrate allocation and consumption of total traffic via APN at the Extra Network Gateways. Describe the bitrate for each Audience Node. Measure the packet loss rate between the Extra Network Gateways (and between the Renderer Node and the Audience Node (optional)) Measure the packet delay between the Extra Network Gateways (and between the Renderer Node and the Audience Node (optional)) Describe measuring method in detail	This is applicable to Advanced Optional Features 2.1.3.2, specifically to Point to Multi-Point.
7	[2.1.5] Other Considerations Provide qualitative and quantitative analysis by comparing IOWN GF technologies with existing technologies.	Described in 8.4, 8.5

Appendix.B Configuration

B.1 Renderer Node and Audience Node

Table B.1-1: Hardware configuration for Renderer Node

Item	Description
Server Platform	Motherboard: ASUSTeK Computer Inc. PRIME H670 PLUS D4
CPU	12th Gen Intel(R) Core(TM) i9-12900 / 1 Socket / 16 Cores
Memory	80GBytes
Storage	Samsung MZ-V8P2T0C/IT SSD PCI-Express 980 PRO (with heatsink 2TB /M.2)
BIOS	Version 1620, 12/08/2022
NIC	NVIDIA ConnectX-5 MCX516A-CCAT
GPU	NVIDIA GA104GL [RTX A4000]
Transceiver	QSFP-SR4 100G (Mellanox compatible)

Table B.1-2: Software configuration for Renderer Node (WebRTC and PoC System (RDMA))

Item	**Description**
Operating System	Ubuntu 22.04, Linux 5.15.0-97-generic
Graphics driver	NVIDIA: 545 CUDA: 12.3
Network driver	rdma-core: 2307mlnx47-1.2310119 rdmacm-utils: 2307mlnx47-1.2310119 librdmacm-dev: 2307mlnx47-1.2310119 librdmacm1: 2307mlnx47-1.2310119
RDMA configuration	Number of Queue Pair to create: 1 Completion Queue monitoring: Start a dedicated thread to poll ibv_get_cq_event() and ibv_poll_cq() for monitoring. Monitoring is not performed every time a Work Request is posted. Parameters for struct ibv_qp_init_attr qp_type = IBV_QPT_RC cap.max_send_wr = 4 cap.max_recv_wr = 4

Table B.1-3: Software configuration for Renderer Node (PoC System (LLMC))

Item	**Description**
Operating System	Windows11 Pro 23H2
Graphics driver	NVIDIA Graphics Driver 552.22, Graphics API: DirectX 12
Network driver	DisplayPort output is being converted, so the PC's Network interface is not being used.

Table B.1-4: Application information for Renderer Node

System	Description
WebRTC Server	Unity 2022.3 + Unity Render Streaming Plugin 3.1.0
PoC System (RDMA) Server	Unity RDMA Plugin, Note: Sony developed for this PoC.
PoC System (LLMC) Server	Unity 2022.3 URP/HDRP project

Table B.1-5: 3D contents for WebRTC System, PoC System (RDMA) and PoC System (LLMC)

Item	Description
A dancer dancing in front of a spinning windmill	Content combining a spinning windmill as a background of Virtual Space and a dancer as an Audience. Windmill and dancer animate based on animation data. Verts: 485K Tris: 506K
Draw 1,000 material by Unity drawMeshInstanced() API	API generated content for verification with reduced rendering load Motion is added by changing the drawing position and color in each frame Verts: 3.7K Tris: 9.1K

Audience Node

Table B.1-5: Hardware configuration for Audience Node (WebRTC System and PoC System (RDMA))

Item	Description
Server Platform	Motherboard: ASUSTeK Computer Inc. PRIME H770-PLUS D4
CPU	13th Gen Intel(R) Core(TM) i9-13900 / 1 Socket / 24 Cores
Memory	16Gbytes
Storage	Samsung MZ-V8P2T0C/IT SSD PCI-Express 980 PRO (with heatsink 2TB /M.2)
BIOS	Version 0808 12/08/2022
NIC	NVIDIA ConnectX-5 MCX516A-CCAT
GPU	NVIDIA GA104GL [RTX A4000]
Transceiver	QSFP-SR4 100G (Mellanox compatible)

Note: Hardware configuration for Audience Node (PoC System (LLMC)) is listed in the “RX” part of Table B.3-1 in Appendix B.3

Table B.1-6: Software configuration for Audience Node (WebRTC and PoC System (RDMA))

Item	Description
Operation System	Ubuntu Server22.04, Linux 5.15.0-91-generic
Graphics driver	NVIDIA: 545 CUDA: 12.3.2
Network driver	rdma-core: 2307mlnx47-1.2310119 rdmacm-utils: 2307mlnx47-1.2310119 librdmacm-dev: 2307mlnx47-1.2310119 librdmacm1: 2307mlnx47-1.2310119

Table B.1-7: Application information for Audience Node

Item	Description
WebRTC Client	Google Chrome Browser
PoC System (RDMA) Client	Native Application
PoC System (LLMC) Client	Embedded Application for LLMC hardware with FPGA

B.2 Open APN and FDN

Table B.2-1: Network Emulator configuration (Joint PoC in Osaki)

Item	Description
Network Emulator	Keysight Network Emulator 3
Transceiver	QSFP-SR4 100G (Mellanox compatible)
Optical Cable	12 fibers OM4 50/125 Multimode Trunk Cable (MTP-MTP) 10m x 2

Table B.2-2: Open APN Hardware information (Joint PoC in Abiko)

Item	Description
APN-T	NEC WX-T(Galileo Flex T)
Transceiver	100GBASELR4 10km, SOURCE PHOTONICS, SPQ-CE-LR-CDFF Complies with 21CFR 1040.10 and 1040.11
Optical Cable	Cable length 25km, 80km

Table B.2-3: Open APN Hardware information (Joint PoC in Osaki)

System	Description
PtP/PtMP Open APN Extra-Network Gateway	Sumitomo Electric FTU9100 Ethernet Switch
Open APN transceiver (APN-T)	Infinera 100Gbps ICE-X coherent transceiver (complies with Open XR Forum and QSFP-DD MSA)
Ethernet interface transceiver	100GBASE-SR4

B.3 Low Latency Media Converter (LLMC)

Table B.3-1: Low Latency Media Converter information (Joint PoC in Osaki)

Item	RX (in Audience Site)	TX (in Renderer Node)
Model	Sumitomo Electric LLMC Prototype (based on FPGA evaluation board)	
FPGA	Xilinx XCKU5P	
WAN interface	100Gbps Ethernet 100GBASE-SR4 (with/ 4 x 25G SFP28 / 1 x QSFP28 Transceiver cable)	
USB interface (for User Control Data)	USB2.0 x 1 input	USB2.0 x 1 output

DisplayPort (DP) interface (for Rendered video)	VESA DisplayPort1.4 x 1 output	VESA DisplayPort1.4 x 1 input
Display Stream Compression	Transparent	

Appendix.C Steps in Calculating Projected Numbers

The values shown in the figure were converted to benchmark model equivalents.

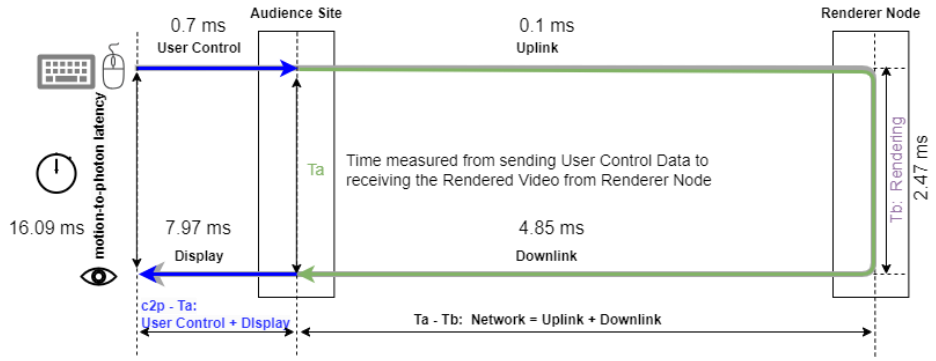

Figure C-1: motion-to-photon latency breakdown

STEP1: Obtain actual measurements of 2K resolution.

- The numbers in the table are actual measured numbers at 2K resolution.

	T_c2p (msec)	Ta (usec)	C2p – Ta (usec)	Tb (usec)	Ta – Tb (usec)
Min	10.33	3647	5828	1047	2592
Max	19.52	6777	4126	4126	2806
Average	14.02	5088	8934	2474	2614

- Ta: 5.08 msec, Ta is the time from when the client receives the User Control data until the video data arrives at the Audience site from the Renderer Node.
- Tb: 2.47 msec, the Rendering time
- Ta-Tb: 2.61 msec, Since Tb is the Rendering time, Ta-Tb means the sum of the communication channel time.

STEP2: Break down communication channel time into uplink / downlink

- The above 2.61 msec is the total time spent on the communication channel, so it is broken down into two parts: uplink and downlink.
- Uplink: 0.10 msec
- Downlink: 2.51 msec

STEP3: Convert the Downlink time to the equivalent of 4K resolution. In addition, calculate the value when bandwidth is lowered by using DSC.

- The 2.51 msec above is for 1920x1080 video data, convert this to 4K video data, where 4K is assumed to be 24-bit and HD is 32-bit.
- 2.51 * (4000*4000*24) / (1920*1080*32)/3 = 4.85 [msec]

- Assuming that there are no restrictions on the 31 Gbps bus width of the PCI of the PC used, this translates to about 1/3 of the time since 100 Gbps of bandwidth is available. This is based on actual measurement know-how at Fujitsu.
- Furthermore, if 1:3 DSC is applied, the data size is 1/3 and 1/3 of the time is 1.62 msec, which is 3.23 msec less.

STEP4：Downlink time for 4K was obtained, add other values

- First, subtract the 2K downlink time from the original Ta, then add the 4K downlink time to obtain the 4K equivalent Ta, 5.08 – 2.51 + 4.85 = 7.42 [msec]
- Add to this the Input latency of the Mouse (Acer ROG KERIS), the average Display output, and the Frame Buffer copy time.
- Mouse 0.7 msec, The time required for the client PC to detect the event after the switch is physically pressed and a click event is issued.
- Display output average 4.17 msec (=1/120*1000/2), The time it takes for a picture to be sent from the client PC to the display and actually displayed, which is half of 1/fps on average
- Frame buffer copy 3.80 msec Current measured value of unoptimized portion of Graphics processing in Linux from Latency decomposition
- 7.42 + 0.7 + 4.17 + 3.80 = 16.09 [msec]

STEP5：Calculate the value if two optimizations are performed

- Since the above is the value for uncompressed images, which is 1/3 of the value for DSC, 16.09 - 3.23 = 12.86 [msec] and shortened
- If we can optimize Graphics from Latency decomposition, we can omit 3.80 msec.
- 12.86 - 3.80 = 9.06 [msec]
- Similarly, if the calculations in STEP 4 are modified to match the Display's display frequency, the estimates at 240 fps and 360 fps can be obtained in the same way. Using the above estimation method, a value for 4K resolution was projected from the actual 2K value.

Appendix.D CPU/GPU Utilization

For CPU usage, we used a tool called Intel PCM and used the numbers in the “C0 core residency” section shown in the red box in the figure.

```
Core (SKT) | EXEC | IPC  | FREQ  | AFREQ | L3MISS | L2MISS | L3HIT | L2HIT | L3MPI | L2MPI |  TEMP

  0    0     0.02   0.65   0.03    1.94    227 K    169 K    0.86    0.71  0.0049  0.0037    61
  1    0     0.03   0.75   0.04    1.94    316 K    197 K    0.85    0.82  0.0043  0.0027    63
  2    0     0.03   0.51   0.06    1.94    540 K    292 K    0.86    0.65  0.0077  0.0042    62
  3    0     0.02   0.76   0.02    1.94    205 K    117 K    0.87    0.69  0.0047  0.0027    62
  4    0     0.02   0.77   0.02    1.93    309 K    126 K    0.87    0.57  0.0072  0.0029    63
  5    0     0.01   0.44   0.02    1.93    130 K    110 K    0.94    0.52  0.0070  0.0059    63
  6    0     0.03   1.02   0.03    1.93    337 K    161 K    0.84    0.76  0.0045  0.0022    63
  7    0     0.02   0.51   0.03    1.94    172 K    191 K    0.94    0.63  0.0046  0.0051    63
  8    0     0.59   2.22   0.27    1.94   2771 K    730 K    0.75    0.72  0.0019  0.0005    64
  9    0     0.11   0.69   0.16    1.94   1947 K    796 K    0.82    0.71  0.0074  0.0030    64
 10    0     0.01   0.57   0.02    1.94    291 K    114 K    0.82    0.75  0.0093  0.0037    64
 11    0     0.12   2.22   0.05    1.95    360 K    182 K    0.84    0.65  0.0013  0.0006    64
 12    0     1.81   3.33   0.55    1.94   1902 K    718 K    0.81    0.72  0.0004  0.0002    61
 13    0     0.15   1.03   0.15    1.94   2010 K    643 K    0.79    0.73  0.0054  0.0017    58
 14    0     0.04   0.96   0.04    1.93    857 K    211 K    0.73    0.69  0.0093  0.0023    64
 15    0     0.03   0.64   0.05    1.94    550 K    263 K    0.82    0.74  0.0068  0.0033    64
 16    0     0.03   1.16   0.03    0.63    563 K     83 K    0.59    0.78  0.0079  0.0012    65
 17    0     0.02   1.26   0.02    0.41    232 K     40 K    0.67    0.82  0.0042  0.0007    65
 18    0     0.04   1.41   0.03    0.30    383 K     74 K    0.74    0.74  0.0041  0.0008    63
 19    0     0.02   0.76   0.02    0.38    814 K    132 K    0.64    0.71  0.0200  0.0033    63
 20    0     0.03   0.67   0.04    1.17   1309 K    509 K    0.75    0.66  0.0181  0.0071    65
 21    0     0.02   0.81   0.03    0.89    294 K     72 K    0.71    0.88  0.0051  0.0013    64
 22    0     0.02   0.80   0.03    0.38    788 K    150 K    0.60    0.78  0.0142  0.0027    64
 23    0     0.04   0.60   0.07    0.34    824 K    268 K    0.76    0.82  0.0081  0.0026    64
---------------------------------------------------------------------------------------------------
 SKT   0     0.14   1.81   0.08    1.30     18 M   6360 K    0.79    0.73  0.0023  0.0008    50
---------------------------------------------------------------------------------------------------
 TOTAL *     0.14   1.81   0.08    1.30     18 M   6360 K    0.79    0.73  0.0023  0.0008    N/A

 Instructions retired: 7911 M : Active cycles: 4371 M : Time (TSC): 2425 Mticks : C0 (active,non-halted) core residency: 5.77 %

 C1 core residency: 64.08 %; C3 core residency: 0.00 %; C6 core residency: 22.85 %; C7 core residency: 7.30 %;
 C0 package residency: 100.00 %; C2 package residency: 0.00 %; C4 package residency: 0.00 %; C6 package residency: 0.00 %;
                          ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
 Core    C-state distribution³00000111111111111111111111111111111111111111111111111666666666666666666777777³
                          ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
                          ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
 Package C-state distribution³000000000000000000000000000000000000000000000000000000000000000000000000000000³
                          ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

 PHYSICAL CORE IPC                 : 1.81 => corresponds to 30.16 % utilization for cores in active state
 Instructions per nominal CPU cycle: 0.14 => corresponds to 2.27 % core utilization over time interval
 SMI count: 0
---------------------------------------------------------------------------------------------------
MEM (GB)->|  READ |  WRITE | CPU energy | PP0 energy | PP1 energy |
---------------------------------------------------------------------------------------------------
 SKT   0     0.00     0.00      32.21        24.32        0.00
---------------------------------------------------------------------------------------------------
```

Figure D-1: Intel PCM log example

For more details, please refer GitHub, Intel Performance Counter Monitor, https://github.com/intel/pcm

The red box on the left shows power consumption and the right shows GPU Utilization.

```
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 545.29.06              Driver Version: 545.29.06    CUDA Version: 12.3     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA RTX A4000               Off | 00000000:01:00.0  On |                  Off |
| 45%   65C    P2              48W / 140W |   1661MiB / 16376MiB |      7%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|    0   N/A  N/A      2040      G   /usr/lib/xorg/Xorg                          154MiB |
|    0   N/A  N/A      2221      G   ...libexec/gnome-remote-desktop-daemon        3MiB |
|    0   N/A  N/A      2271      G   /usr/bin/gnome-shell                        169MiB |
|    0   N/A  N/A    163216      G   gnome-control-center                          3MiB |
|    0   N/A  N/A    165248    C+G   ..._0402/UnityApp_4cam_20240321.x86_64     1256MiB |
+---------------------------------------------------------------------------------------+
```

Figure D-2: NVIDIA nvidia-smi log example

For more details, please refer the System Management Interface SMI page, https://developer.nvidia.com/system-management-interface

Trademarks

This document may contain references to trademarks or registered trademarks of the following companies: NVIDIA, Intel, Unity Technologies, NEC, Fujitsu, Keysight Technologies, Infinera, Tektronix, Teledyne LeCroy, Sumitomo Electric, VESA and Sony, as applicable. All trademarks mentioned herein are the property of their respective owners.

History

Revision	Release Date	Summary of Changes
0.1	2024/09/12	Initial Draft
0.2	2024/10/08	This version reflects the comments of the RIM TF.
1.0	2024/12/13	This version reflects the comments of the experts.