Design and Implementation of Ultra-realistic Application Service Testbed Supporting End-to-End Ultra-high Precision Networking Architecture

ByeongOk Kwak
Communication & Media Research Lab.
ETRI (Electronics and Telecommunications Research Institute)
Daejeon, Republic of Korea
kwakbok@etri.re.kr

HoSun Yoon
Communication & Media Research Lab.
ETRI (Electronics and Telecommunications Research Institute)
Daejeon, Republic of Korea
yhs@etri.re.kr

Seong Moon
Communication & Media Research Lab.
ETRI (Electronics and Telecommunications Research Institute)
Daejeon, Republic of Korea
seong@etri.re.kr

PyungKoo Park
Communication & Media Research Lab.
ETRI (Electronics and Telecommunications Research Institute)
Daejeon, Republic of Korea
parkpk@etri.re.kr

SeungWoo Hong
Communication & Media Research Lab.
ETRI (Electronics and Telecommunications Research Institute)
Daejeon, Republic of Korea
swhong@etri.re.kr

***Abstract*—The ultra-realistic application service testbed aims to test and verify the core technology of ultra-low latency/ultra-precision network system by applying an ultra-low latency transmission protocol that supports an end-to-end ultra-precision ultra-low latency networking structure to ultra-realistic application services. In this paper, we implemented and proposed an ultra-realistic application service testbed for performance analysis and evaluation of an end-to-end ultra-precision/ultra-low latency networking structure.**

Keywords—ultra-precision network, latency, testbed

I. Introduction

After the pandemic, the transition to a non-face-to-face society has become the main trend in our society, and related technologies have also developed by leaps and bounds. With the development of ultra-realistic technologies that have been noted as key technologies for a non-face-to-face society, various immersive and interactive technologies such as AR(Augmented Reality), VR(Virtual Reality), Metaverse, and HTC(Holographic-Type Communication) have been developed. Supporting ultra-realistic application services are appearing [1, 2]. However, these ultra-realistic application services place significant requirements on networking infrastructure. These include ultra-low latency, high precision, high bandwidth and support for multiple dynamically changing data streams [3, 4]. In particular, delay is very important for ultra-realistic application services.

In this paper, performance analysis and evaluation of end-to-end ultra-low latency transmission protocol and application optimization technology, end-to-end ultra-precision network technology that can minimize the delay occurring in the NIC, networking operating system, and transmission protocol inside the application service processing end [5], are performed. An ultra-realistic application service testbed was implemented to support it.

II. Ultra-realistic Application Service Optimization Test Environment

Metaverse commonly refers to a three-dimensional virtual world and refers to a service that controls a virtual space that can communicate with an avatar that expresses an individual in a virtual space. The service supported by the hyper-realistic application service optimization test environment simultaneously supports avatars and real-time 3D volumetric live images in virtual space.

The scenarios configured in the ultra-realistic application service optimization test environment are as follows. To provide Metaverse-based virtual space service, ETRI protocol is supported to operate in Unity environment.

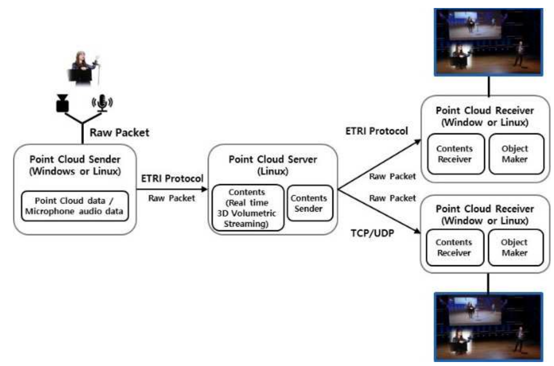

Fig. 1. Scenarios for ultra-realistic application service optimization test environment configuration.

The basic scenario is a service in which virtual avatars and real-user 3D images simultaneously provide remote meetings and seminars in a virtual space. Include the following detailed features.

- Point cloud capture function using depth camera

- 3D volumetric rendering function using Point Cloud

- Real-time 3D volumetric streaming transmission function

- Metaverse-based virtual space composition

- Avatar and real-time 3D volumetric streaming display in virtual space and real-time voice synchronization function

By using Microsoft's Azure Kinect V2 device used in the test environment [6], Depth, RGBA, etc. required for Point Cloud Data creation can be imported. In the Unity environment [7], a 3D volumetric rendering function was implemented using point cloud information (Vector3, RGBA) collected through depth cameras such as Azure Kinect.

In case of streaming function, it can have "Point Cloud Sender", "Point Cloud Server", and "Point Cloud Receiver" structure as shown below, and the role of each module is as follows.

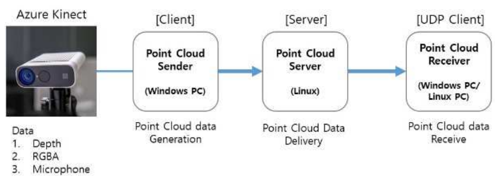

Fig. 2. Point Cloud Streaming Transmission Structure.

- Point Cloud Sender: Receives information (Depth, RGBA, Micro Phone) required to create Point Cloud Data from Azure Kinect, creates and transmits Point Cloud Data (PCD, Point Cloud Data) packets

- Point Cloud Server: Deliver PCD received from Point Cloud Sender to connected Point Cloud Receiver

- Point Cloud Receiver: Create volumetric object with PCD information received from Point Cloud Server

III. Ultra-Realistic Application Service Operation Scenario

The service functions and operating scenarios of each component to support the ultra-realistic application service test environment are as follows.

A. Point Cloud Sender

As for service functions, it has Azure Kinect linkage, Microphone linkage, UDP packetization & transmission function.

1) Integration with Azure Kienct

This is a function that the user connects to Azure Kinect through the Point Cloud Sender, and the order is as follows.

① Select the desired camera information (Depth Mode, FPS) in the GUI.

② Connect to Azure Kinect and start Capture to convert image data into Point Cloud and express it.

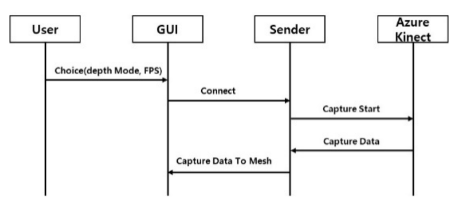

Fig. 3. Block diagram of Azure Kinect integration

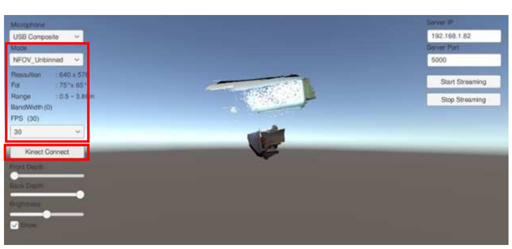

Fig. 4. Azure Kinect Integration GUI

If you select the Kinect Connect button at the bottom of Figure 4, a Point Cloud is created in conjunction with Azure Kinect.

2) Microphone linkage

This is a function that the user connects to the microphone through the Point Cloud Sender, and the order is as follows.

① Select the desired microphone from the GUI.

② Connect to Microphone and start recording to get audio data in real time.

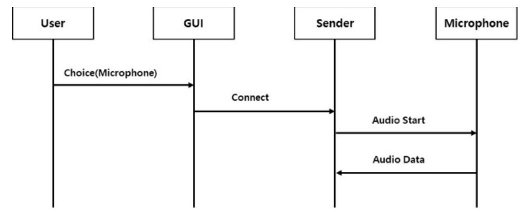

Fig. 5. Microphone linkage block diagram

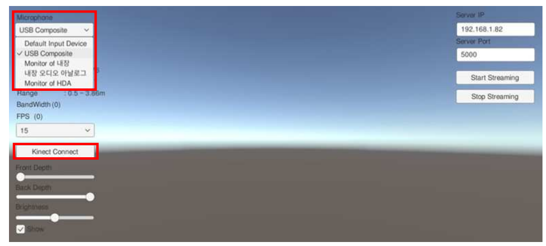

Fig. 6. Microphone linkage GUI

If you select the Kinect Connect button at the bottom of Figure 6, recording starts in conjunction with the Microphone.

3) UDP transmission

This function transmits the data received from the Azure Kinect & Microphone by the user to the Point Cloud Server-er through the Point Cloud Sender. The order is as follows.

① Enter the IP and port number of the server you want to connect to in the GUI.

② Capture data of Azure Kinect and Audio data of Microphone are converted into UDP packets and transmitted through ETRI protocol library.

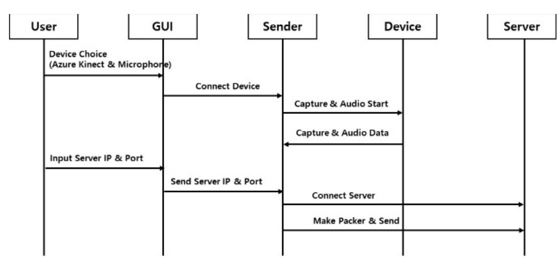

Fig. 7. UDP Packetization & Transmission Block Diagram

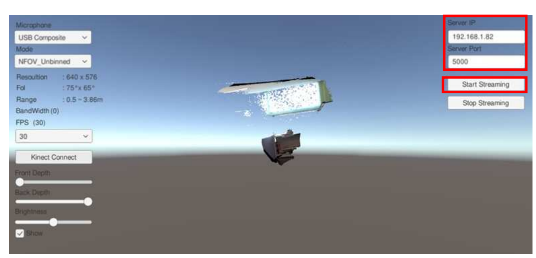

Fig. 8. UDP Packetization & Transmission GUI

After linking Azure Kinect and Microphone, enter the Server IP and Server Port information on the right side of Figure 8 and select the Start Streaming button at the bottom to transmit data through UDP communication.

B. Point Cloud Server

Point Cloud packets received through Point Cloud Sender are transmitted to N Point Cloud Receivers.

① Set the receiving and transmitting ports

② Enter the UDP Client address to be transmitted.

③ Press the START button to operate the server.

④ At this time, the packet is received using the ETRI ultra-low latency transmission protocol API.

⑤ At this time, packet transmission is transmitted using the ETRI ultra-low latency transmission protocol API.

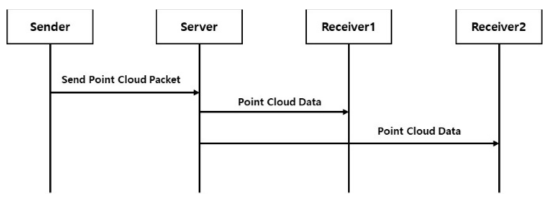

Fig. 9. Block diagram of data transmission

C. Point Cloud Receiver

Point Cloud data received from Point Cloud Server is created as Point Cloud Object.

Point Cloud Receiver receives Point Cloud Packet through Point Cloud Server

① Enter the UDP port number to be received.

② Press the Receive button

③ Check the Point Cloud Object created through the received Point Cloud packet

④ At this time, the packet is received using the ETRI ultra-low latency transmission protocol API.

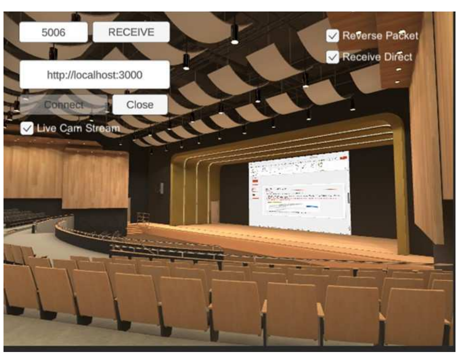

Fig. 10. Point Cloud Receiver GUI

On the Point Cloud Receiver screen in Figure 10, check if the Point Cloud Object is the same as the one sent by the sender.

For functional and performance tests, NFOV Unbinned Mode was used in this paper. In the UDP test, after dividing one frame data into 1,400 bytes size, RTP header was created in each packet and transmitted through UDP (approximately 2,634 packets). In the TCP test, one frame data was transmitted (approximately 3,686,400 Bytes) through TCP at one time.

When 30 frames are transmitted, almost 900 Mbps of data including the header must be transmitted. In the case of TCP, the packet is adjusted and sent according to the buffer situation of the receiving side, but in the case of UDP transmission, the buffer situation of the receiving side is not considered. In the case of UDP transmission, it can be seen that packets are transmitted according to network conditions, causing network load and packet loss.

In the case of UDP transmission, the burden of dividing one frame data to create and attach the RTP header occurred, and the load of restoring the frame data again occurred on the receiving side.

In the case of the receiving server in the test environment, it was confirmed that the I/O load was higher than that of the transmitting program or the receiving program while receiving and transmitting data of about 1Gbps from one network port.

IV. CONCLUSION

In this paper, an ultra-realistic application service testbed was implemented to analyze and evaluate the performance of an end-to-end ultra-precision/ultra-low latency networking structure. The ultra-realistic application service testbed can be used for high-precision real-time operating systems of industrial terminals that require ultra-low latency and high reliability, such as V2X and remote control, and ultra-low latency and high-capacity server systems that realize ultra-realistic content services such as XR and hologram.

In the future, by applying ultra-low latency transmission protocol and ultra-low-latency network operating system that supports end-to-end ultra-precision/ultra-low latency networking structure to ultra-realistic application service testbed, core technology test and It will be used for verification.

ACKNOWLEDGMENT

This research was supported by the IT R&D program of MSIT (Ministry of Science and ICT) [2021-0-00715, Development of End-to-End Ultra-high Precision Network Technologies]

REFERENCES

[1] D. Fox; I. G. Thornton, "IEEE White Papers," in The IEEE Global Initiative on Ethics of Extended Reality (XR) Report—Extended Reality (XR) Ethics and Diversity, Inclusion, and Accessibility , vol., no., pp.1-25, 4 March 2022.

[2] McGill; Mark, "The IEEE Global Initiative on Ethics of Extended Reality (XR) Report--Extended Reality (XR) and the Erosion of Anonymity and Privacy," in Extended Reality (XR) and the Erosion of Anonymity and Privacy - White Paper , vol., no., pp.1-24, 18 Nov. 2021.

[3] N. H. Mahmood, H. Alves, O. A. López, M. Shehab, D. P. M. Osorio and M. Latva-Aho, "Six Key Features of Machine Type Communication in 6G," 2020 2nd 6G Wireless Summit (6G SUMMIT), 2020, pp. 1-5, doi: 10.1109/6GSUMMIT49458.2020.9083794.

[4] V. Ziegler and S. Yrjölä, "How To Make 6G a General Purpose Technology: Prerequisites and value creation paradigm shift," 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), 2021, pp. 586-591, doi: 10.1109/ EuCNC/6GSummit51104.2021.9482431.

[5] B. Kwak, H. Yoon, S. Moon, P. Park, S, Hong, "Design for End-to-End Ultra-high Precision Networking Architecture," International Conference on ICT Convergence 2022 (ICTC2022), pp.2359-2361, Oct 2022.

[6] https://learn.microsoft.com/ko-kr/azure/kinect-dk/

[7] https://unity.com/kr