16th Int. Conf. on Accelerator and Large Experimental Control Systems ICALEPCS2017, Barcelona, Spain JACoW Publishing
ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA040

DEVELOPMENT OF REAL-TIME DATA PUBLISH AND SUBSCRIBE SYSTEM BASED ON FAST RTPS FOR IMAGE DATA TRANSMISSION*

Giil Kwon[†],Jinseop Park, Tae Gu Lee, Taehyun Tak, Woongryol Lee, Jaesic Hong,
National Fusion Research Institue(NFRI), Deajeon, Republic of Korea

Abstract

In fusion experiment, real-time network is essential to control plasma real-time network used to transfer the diagnostic data from diagnostic device and command data from PCS(Plasma Control System). Among the data, transmitting image data from diagnostic system to other system in real-time is difficult than other type of data. Because, image has larger data size than other type of data. To transmit the images, it need to have high throughput and best-effort property. And To transmit the data in real-time manner, the network need to has low-latency. RTPS(Real Time Publish Subscribe) is reliable and has Quality of Service properties to enable best effort protocol. In this paper, eProsima Fast RTPS was used to implement RTPS based real-time network. Fast RTPS has low latency, high throughput and enable to best-effort and reliable publish and subscribe communication for real-time application via standard Ethernet network. This paper evaluates Fast RTPS and Zstd about suitability to real-time image data transmission system. To evaluate performance of Fast RTPS and Zstd base system, Publisher system publish image data and multi subscriber system subscribe image data.

INTRODUCTION

KSTAR is a superconductive tokamak fusion reactor in Korea. KSTAR uses image data to analyse plasma status and protect device. Interlock system can use thermal imaging camera to monitor tokamak wall temperature status and to protect device. And TV Image Diagnostic system uses image data to analyse and understand status of plasma. In KSTAR, raw image data is required to transmit multiple remote server in real time. Because Image data acquired from image DAQ system cannot process in real time in DAQ system. and must be store to archiving server. And some diagnostic image data need to monitored in real time. In KSTAR, image DAQ system transfer image data to processing server. Because Processing image data take much computing cost. If image data processed in DAQ system, it can interfere the DAQ process. Therefore, we distribute system to image DAQ system and image processing system. most of the image are connected with Ethernet and image data transmitted by manual. we try to make automatic image processing routine by using the real-time image transmission system. by using these system, we can simplify image data sharing among servers which needs to image data in real time.

Transferring the image data in real time is challenging problem. Because, most of image data size is larger than other type of data.

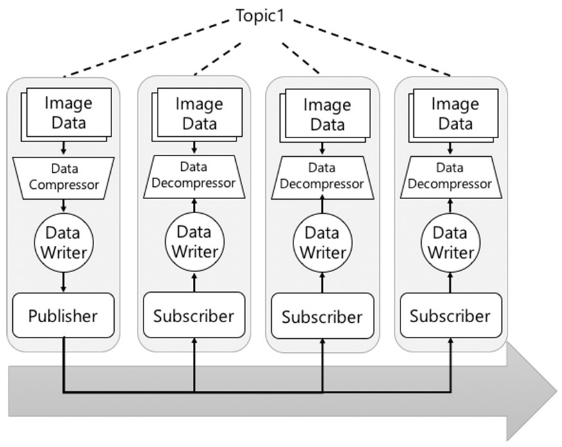

Figure 1: Configuration of image transmission system.

Sending/Receiving image data through network will takes most of network bandwidth and slow down the network performance. and publisher/subscriber node's network latency also will be degraded.

One image source required to transfer to multiple destination. For example, Image DAQ server acquires images from camera and simultaneously send this data to archiving server, image processing server and monitoring server. Increasing number of receiving server can create heavy load on data sending server. We use Real Time Publication Subscribe Protocol(RTPS) to reduce load to publish node and effectively share the image data. As figure 1 show us, real time image transmission system consists of compression/decompression module and RTPS module. Image data compress with lossless compress algorithm and publish to multiple subscriber. Each subscriber subscribe data and decompress data to get image data.

DESIGN AND IMPLEMENTATION OF THE IMAGE TRANSMISSION SYSTEM

Real Time Publication Subscribe Protocol(RTPS)

RTPS is a publish/subscribe protocol for Data Distribution Service(DDS) implementations. DDS is a network communication middleware Object Management Group(OMG) standard. RTPS facilitates scalable, real-time, reliable and high-performance system. RTPS provides best effort and reliable QoS reliability mode. By configuring the system to best effort QoS mode, we can implement high performance networking system. RTPS implements publisher/subscriber pattern to simplifies complex network programming.

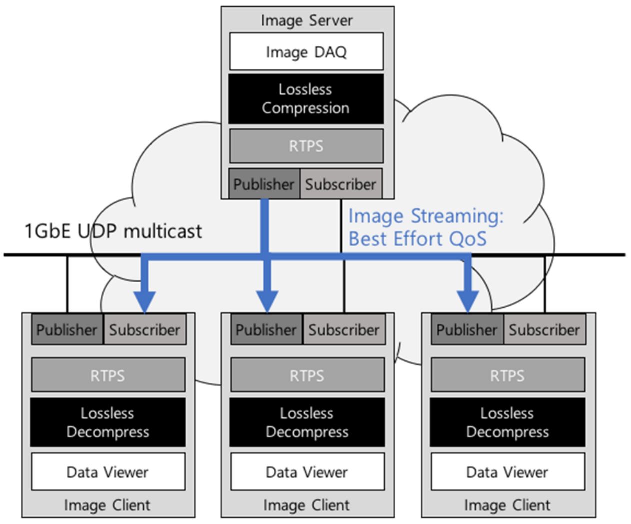

Figure 2: Configuration of RTPS and lossless compression algorithm based real time image transmission system.

. RTPS uses this pattern to provide sending/receiving function for events and data among nodes. Each node which share topic can easily share data as if each node were one server. For this functionality, RTPS automatically discover the address of nodes which has same topic. Each node does not require setting address in advance. KSTAR implement real-time image transmission based on RTPS. Image DAQ node acquires image from camera and publish data with specific topic. And multiple image Clients subscribe this data. (such as image archiving server, image processing server). Each node publish/subscribe image data over 1GbE UDP multicast with best effort QoS reliability mode. To implement RTPS based system, we use eProsima Fast RTPS framework[1]. Fast RTPS is a fast publication/subscribe framework. Fast RTPS compliant with OMG RTPS 2.2 compliant. And this framework compatible with other RTPS implementations (such as RTI Connext DDS, OpenSplice DDS, and Core DX). This support multiple Operation System (such as linux, windows, mac os).

Lossless Real Time Compression Algorithm

Compressing image data is essential for real-time image transmission. Because image data size is too large to transfer image through network. Our system use real time compression method to compress image data. Real time compression method compress and decompress data in real time. Generally, video codec is widely used to reduce the size of image (such as H.264, VP6). Dettie et el[2], Mohammed et el[3] and B Al-Madani at el[4] proposed video streaming over RTPS. They encode video stream with H.264 codec. But most of codec are lossy methods. The encode and decode image does not same to original image. In fusion research, diagnostic data integrity is important. If data changed while transferring to other system, the accuracy of diagnostic analysis will be degraded. Therefore, lossless compress is important for this system. there is some kind of lossless video codec. Lossless encoding codec guarantee mathematical lossless. However, while converting image format (such as RGB to YUV 4:4:4 conversion), image data loss may occur by data type conversion. To ensuring the image data integrity, we compress data with lossless compression algorithm. In this system, we use Zstandard(zstd)[5] algorithm to compress image data. Zstd is real time compression algorithm. This algorithm is fast and provide high compression rate. Zstd has dictionary mode that can train sample data and compress with this train data. This mode is good for the case when the compressed data is small(<MB). Dictionary mode compression is more faster and higher compression rate than standard mod. Our system also configured to used dictionary mode. Before transferring image data, we use 3~5 image to train dictionary. By using this dictionary, we compress each image coming from camera.

Image Transmission Process

Figure 3 represent the process of real time image transmission process. Real time image transmission process is following as below:

- Image publisher node acquire image from camera.
- Real time compression method train image to make dictionary for the compression (This process is performed only once at the beginning.)
- Compressed data are published to subscriber.
- Subscriber nodes subscribes image data and decompress the data.

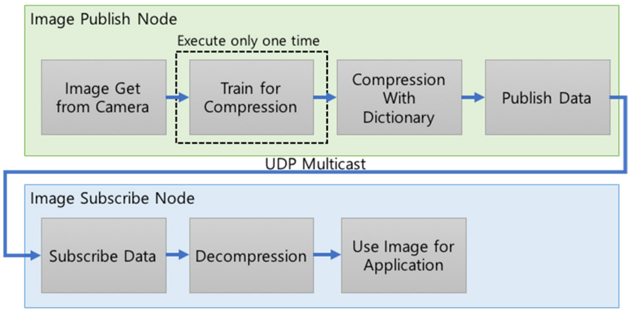

Figure 3: Real time image transmission process.

PERFORMANCE TEST AND RESULT

To validate performance of this system, we did loopback latency test while vary image resolution. The test environment is following as bellow:

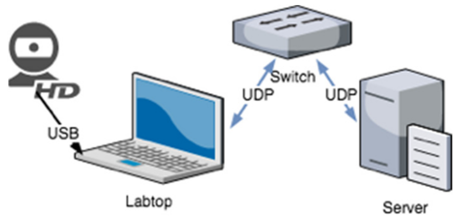

Figure 4: Test environments.

Table 1: Hardware Specifications

Margin	Node 1	Node2
CPU	Intel Core i7 (2.7GHz, 4core)	Intel Core i3 (3.3GHz, 2core)
Memory	16GB	8GB
Storage	1TB SSD	8GB

Two computer was used to do performance test. Two nodes are connected with 1GbE.

Loopback Test

Loopback test is the process which measure the latency of network by calculating the time difference between sending packet time and receiving packet time from source node.

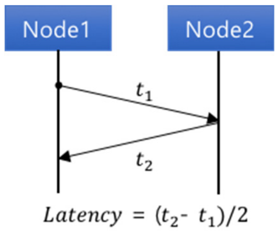

Figure 5: Concept of loopback test.

Node1 acquire image from camera and publish data with the topic that has publish time information and image data. When node2 subscribe the data, node2 publish with different topic that has received image data and publish time information. Latency is the time difference between publish time and returned topic subscribe time. For more accurate test, we conducted 500 times test and compute the statistics of test data.

Figure 7 show us the process of loopback test between 2 nodes.

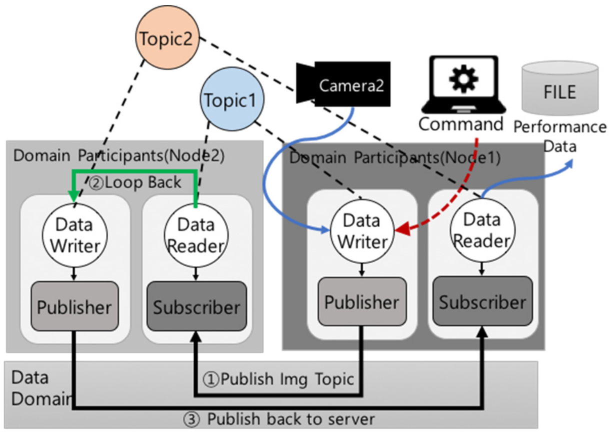

Figure 6: Concept of loopback test.

Node1 starts the test with user command and publish image data with topic1 and node2 make the topic2 packet with image data received from node1. And publish time information. Node2 publishes this data and node1 subscribe topic2 packet. After subscribe topic2 packet, node 1 compute the latency of network. Node1 store every time spent on each task and image data size into memory and after finishing performance test. Node1 write the data into file system.

Experiment Results

We did performance test while varying image resolution. As Table 2 show us, we test the image which has 16:9 aspect ratio. We test the color and monocular image.

We did not experiment with 1280x720 color image.

Table 2: Image Resolution and Size

Resolution	Gray image size(KB)	Color image size(KB)
160x90	14.06	42.18
320x180	56.25	168.75
640x360	225	675
1280x720	900	2700

As Figure 7 show use, Node1 store every time spent on each process. Stored process time is following as bellow:

Table 3: Time Consumed to Process 640x360 Gray Image

Name	Time(msec)
Training for compression	9978
Image acquisition	47.82
Compression	19.49
Network Latency	6.18
Decompression	4.97

And we calculate the percentage of time consumed for each process.

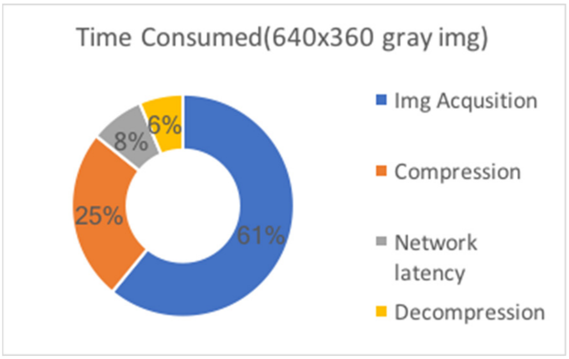

Figure 7: Percentage of time consumed by each process when 640x380 gray image transmitted to subscriber nodes from publisher node.

Training dictionary for compression take almost 9 sec when training 640x360 gray image. This process take most of process time. This process run at once at first time. Image acquisition process take 60% of total process time. This time dependent to camera device and driver. Therefore, this time can be changes as camera changes. Compression time take 25% time. But this time are depended to image size. As Figure 8 show us, as the image resolution increases, compressed image size decreases. Minimum compression rate of the system is 80%.

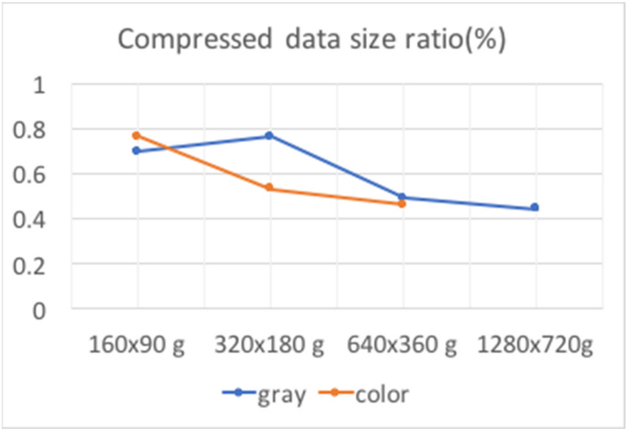

Figure 8: Percentage of compressed data size.

Network latency was calculated by subtracts the topic1 packet publish time from node1 and topic2 subscribe time from node 1. And then we divide this value with 2.

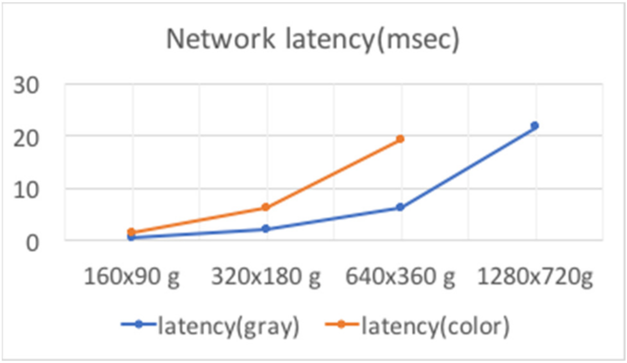

Figure 9: Network latency.

As Figure 9 show us, as image size is increases, latency also increases. 640x360 grey image take 6msec latency. If we transmit 640x360 grey image over 1GbE UDP network, it will take 6msec and its frame rate per second(fps) will be almost 166.

Decompress process is faster than compress process. It takes about 1/4 of the time. Compress and Decompress process time are getting larger as the image resolution increases. Especially compression time exponentially increases as the image size increases. Figure 10. And Figure 11 show us the time consumed by compression process and decompression process.

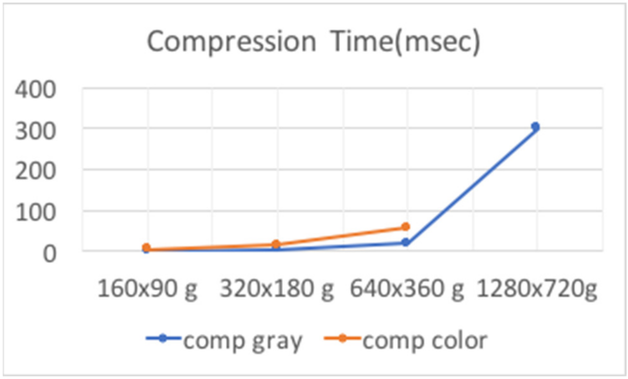

Figure 10: Compression time.

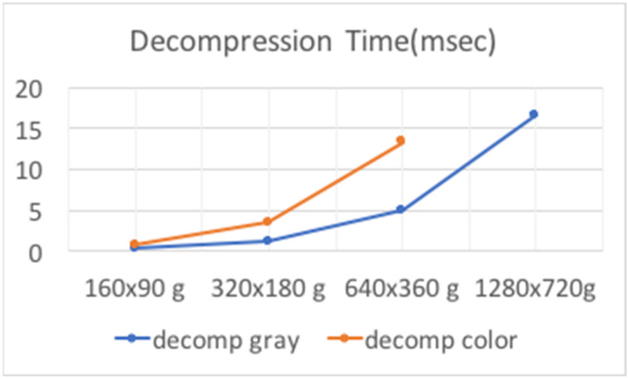

Figure 11: Decompression time.

We also calculate the frame rate per second(FPS) of publisher nodes. Fps decreases as the image size increases. And fps converges to 30fps. Because the camera used for performance test only support 30fps. If we change the camera we can get more fps. The fps of 640x360 gray image is 15.

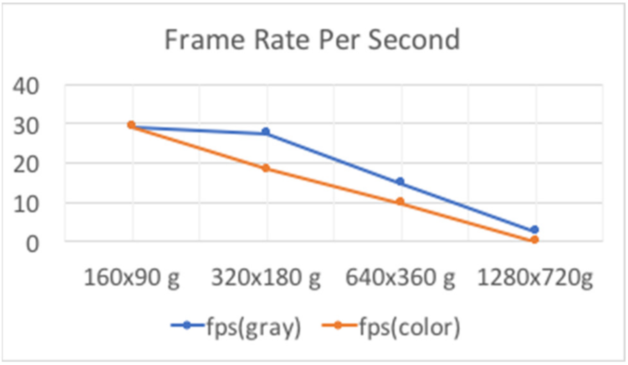

Figure 12: Decompression time.

Compression take time however this process reduces image data size to minimum 80%. By compressing the data, we can reduce the transmission time. As the image size is increase, we can get more benefit from compression.

CONCLUSION

In this paper, The real-time Image transmission system has been developed transfer image data to destination nodes in real time. A real-time Image transmission system was developed by using Fast RTPS and Compression library(Zstd). Publisher nodes send data to multi subscriber nodes without additional computational cost. Because RTPS can configured to run over UDP multicast. The system do lossless compression to acquired image from camera to reduce size of data and network latency (minimum compressed data size is 80% of original size, network latency is 6msec per 640x360 monocular image, compression take 20msec per image). The publisher node run at 3fps (1280x720 gray image), 15fps (640x360 gray image), 27fps (320x180 gray image). We will do more test with 10GbE network environment. In this network, our system performance will be improved. And we also plan to development h.264 codec based image transmission system for remote image viewing service. Despite the fact that h.264 codec will be minutely damage the original image it will make visually lossless image.

ACKNOWLEDGEMENT

This work was and performed within the cooperation defined in the Memorandum of Understanding (MoU) between ITER International Organization (IO) and NFRI and was partially supported by the Korean Ministry of Science ICT & Future Planning under the KSTAR project.

REFERENCES

[1] eProsima Fast RTPS, http://www.eprosima.com/index.php/productsall/eprosima-fast-rtps
[2] Detti, Andrea, *et al.*, "Streaming H. 264 scalable video over data distribution service in a wireless environment", World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a. IEEE, 2010.
[3] AL-MADANI, Basem; AL-SAEEDI, Mohammed; AL-ROUBAIEY, Anas A., Scalable wireless video streaming over real-time publish subscribe protocol (rtps). In: Distributed Simulation and Real Time Applications (DS-RT), 2013 IEEE/ACM 17th International Symposium on. IEEE, 2013. p. 221-230.
[4] Al-Madani, Basem, Anas Al-Roubaiey, and Zubair A. Baig, "Real-time QoS-aware video streaming: a comparative and experimental study", Advances in Multimedia 2014 (2014): 1.
[5] Zstandard, http://facebook.github.io/zstd/