A Stream Merge Method to Reduce Load for Sensor Data Stream Delivery

Ei Khaing Win[1], Tomoki Yoshihisa[1], Ishi Yoshimasa[1], Tomoya Kawakami[2], Yuuichi Teranishi[3, 1], Shinji Shimojo[1]

[1] Osaka University
Ibaraki, Osaka, Japan 567-0047

[2] Nara Institute of Science and Technology
Ikoma, Nara, Japan 630-0101

[3]National Institute of Information and Communications Technology
Koganei, Tokyo, Japan 184-8795

Abstract—Sensor data stream delivery technology takes an important role for the applications. As the number of the streams increases, the server's load caused by the delivery such as the computational power, memory usage, network bandwidth, etc. increases. However, there has never been a method to reduce the number of the streams for sensor data stream delivery. Hence, we propose a stream merge method to reduce the server's load for sensor data stream delivery. In the demonstration, we will show how our proposed method reduces the number of the streams by visualizing various communication situations.

Keywords—sensor data stream delivery; IoT (Internet of Things); streaming distribution; sensor network

I. INTRODUCTION

Due to the recent proliferation of sensors such as cameras or temperature sensors, sensor data stream delivery technology takes an important role for the applications. For example, a smartphone requests a live camera stream to the delivery server and the user checks the current destination situation. In sensor data stream delivery, the number of the streams affects the server's load such as computational power, memory usage, network bandwidth, etc. In the following cases, the number of the streams increases.

- A client requests a live camera stream to show the current situation of the area covered by the camera. Another client requests the same data but the past-recorded video to show the previous situation. In this case, the server delivers two different streams although the video contents itself is the same.
- Some clients (smartphones) receive a real time temperature data stream to check the climate of the travel destination. One of them moves to underground and losses some of the temperature data since the client cannot catch electromagnetic waves in underground. So, the client requests the stream for past data to recover the lacked data. In this case, the server delivers two different streams although the contents of the data stream are the same.

In the above cases, reducing the number of the streams can relieve the server's load. Some methods to reduce the number of the data streams have been proposed in ([1]-[3]). In the sensor data stream delivery, the clients do not always request the data from the beginning, i.e., the start of obtaining sensor data. There has never been a method to realize this for sensor data stream delivery. Hence, we propose a stream merge methods for the cases that the server delivers some sensor data streams but some of them are the past data. The server's load can be relieved by merging some streams since the number of the streams decreases. In the demonstration, we will show how our proposed method reduces the number of the streams.

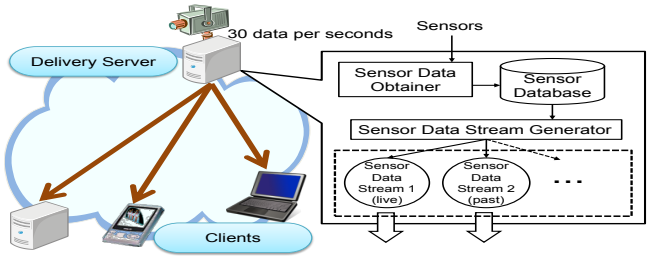

Fig. 1. Assumed System Model

II. PROPOSED METHOD

In this section, we explain our proposed method.

A. Assumed System Model

Figure 1 shows our assumed system model. A sensor is connected to a delivery server. The server obtains sensor data from the connected sensor cyclically and stores the data to its storage. In the figure, the sensor data obtainer obtains the images from the connected camera. The data rate is 30 frames per seconds. Clients request the sensor data stream delivery to the server. Normally, the server generates the real time sensor data stream and delivers it to the requested client. The data rate is the same as that of the sensor data obtainer and is 30 frames per seconds in the case of the figure. If the client requests past sensor data stream, the server generates the stream using the stored data and delivers it to the client. The maximum data rate is the same as that of the sensor data obtainer, but the data rates for past data are controllable.

Examples for actual systems are delivering a live camera stream to some smartphones or delivering sensor data stream to some computational machines.

As explained in Section 1, the server's load increases as the number of the streams increases. So, our proposed method reduces the number of the streams indicated by the dashed square in the figure and reduces the server's load.