SLiM : 스케일러블 라이브 미디어 스트리밍 프레임워크

류 은 석*　　유 혁**

◈ 목 차 ◈

1. Introduction

Rapid developments in the mobile handset and networking technologies are expediting the advent of the ubiquitous age. At the current rate, the homes and offices of the future will evolve to form a U-City that showcases the entire city with ubiquitous devices and sensors. A key piece to realize U-City is the network middleware technology for encoding and streaming to various types of terminals the multimedia data acquired by a number of camera arrays and sensors installed in multiple locations throughout the city [1, 2]. This paper investigates the technical difficulties associated with the large-scale media streaming and proposes solutions for overcoming the difficulties at the middleware level. The proposed framework focuses on improving the traditional media streaming system by streaming high-capacity scalable live media and developing it into practical applications.

There are many problems in streaming large scale live media. The first step for overcoming such difficulties is to identify the problems and then find the solutions. In this section we define the problems and give ideas for initial solutions.

The conventional small-scale media streaming technique shown in Figure 1 involves delivering the multimedia data captured by the camera to the server, encoding it and delivering it to the specified clients with an identical capability. Such approach is difficult to apply in large-scale service environment that contains hundreds or thousands of cameras, supports diverse user terminals and requires multiple simultaneous user interactions.

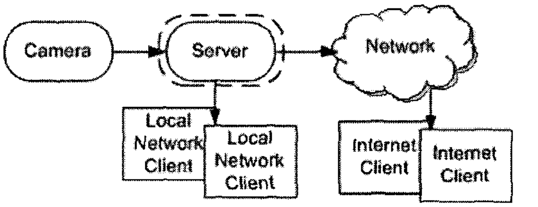


Fig. 1. Conventional Media Streaming System Model

1.1. Towards building large scale live media streaming system

The main technical difficulties for applying a traditional streaming system to a city-wide multimedia delivery system are as follows.

1. Hotspot Problem
2. Camera Array Management
3. Media Streaming Management

* 고려대학교 컴퓨터학과 박사과정(수료)
** 고려대학교 컴퓨터학과 교수

First, let's consider a system for controlling the traffic flow or providing multimedia tourist information using the cameras installed throughout a city. Since the locations of people's interest are limited to a few, data requests will be concentrated on the multimedia data from those spots. In order to manage such "hotspot" requests for specific resources, transcoding, scalable encoding and resource management are needed. Moreover, the camera array management module is necessary to handle operation of the numerous cameras that can be added or removed in real-time. In addition, media stream management is required for variable user device support and the multimedia services capable of coping with the network changes as well as processing the service request for supporting user interaction. However, simply expanding the traditional system to resolve such problems will tremendously increase the server load, as indicated with a dotted line in Figure 1, so making it difficult for the camera module to conduct real-time processing. In other words, a bottleneck will be at the camera server. An obvious solution is simply increasing the number of servers and constructing a SAN(Storage Area Network), but there is a need for analyzing the amount of server processing load in details and systemizing the server roles according to the analyzed server loads. This paper proposes a method for dividing camera management and media stream management and implementing a two-step processing technique according to the resource request amount, as shown in Figure 2. It shows a conceptual division of task among servers. A mechanism is also presented for managing the camera arrays and the media stream. In turn, this paper explains a live media streaming system of the middleware level that improves the conventional multimedia streaming approach.

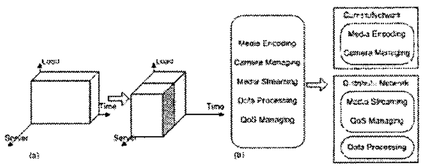


Fig. 2. Conceptual division of tasks among servers

From a conceptual point of view, if the number of servers is fixed, the higher the server load gets, the longer the processing time will be, as shown in Figure 2-a. Therefore, when there is a high quantity of multimedia data from multiple cameras, a conventional system is unable to stream real-time live media. When the roles are allocated to a respective server as shown in the figure on the right of Figure 2-a, the processing time for the task load is reduced. Such approach is different from simply increasing the number of servers with an identical role, and it facilitates the QoS(Quality of Service) management, including server allocation according to the overall system resources. Furthermore, when using multimedia data from cameras for different services such as traffic flow control, security monitoring, video tourist information and disaster measure system, dividing the acquisition, processing and distribution stages for the multimedia data enables easy deployment in various application systems.

In order to take on the roles to handle a large number of cameras described in this paper, the task unit of each server is shown in Figure 2-b. It shows server tasks are classified into 2-step according to a role. A server must manage multiple camera arrays, which must acquire data from the cameras and perform multimedia encoding using a necessary codec. Moreover, if there is a need to analyze the media data, the server must manage the QoS for analysis, processing and delivery, based on which media streaming is provided. These main categories of tasks are classified into two major stages in this paper according to the importance of the roles as well as the process flow.

The first stage is the camera network and the second is the distribution network. The camera network is the stage responsible for camera array management as well as encoding the multimedia data provided by the cameras. The distribution network plays the role of efficiently distributing the multimedia data acquired by the cameras. Each stage is explained in detail in the main chapters of this paper.

Figure 3 displays the conceptual design of the overall framework obtained by dividing the process task into two stages described above. The Scalable Live Media Server(SLiM Server: Camera Server role) under development manages and controls multiple cameras and encodes multimedia data acquired by the cameras. A scalable encoding technique is used in this stage. The streaming server streams the encoded multimedia content according to the QoS. Moreover, at this stage, the distributing server transfers the media to the corresponding data processing server according to the registered information, where it is analyzed, processed and fabricated. For example, in the case of the traffic flow control system, the multimedia data is recognized and analyzed at this stage to determine the traffic flow control, notifies the traffic light processing system to provide control commands.

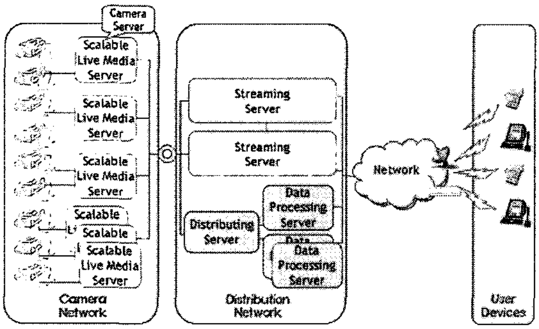


Fig. 3. High Level Conceptual View of the SLiM

This paper is structured as follows: Section 2describes the system architecture and the research progress. Section 2.1 and 2.2 explain the camera array management techniques and the media stream distribution techniques. Finally Chapter 3provides the conclusion of this study as well as potential topics for future studies.

2. System Architecture

The SLiM system is designed to be a middleware responsible for processing and delivering multimedia data captured by the cameras of the ubiquitous city. As explained earlier, the system consists of the camera network and the distribution network. The camera network manages the camera arrays and encodes the video obtained by the cameras. The distribution network receives the multimedia data from the camera network and transmits it to the application server or streams it to the user device. An example of an application server is a traffic flow control system. As indicated in Figure 4, the system receives the multimedia data from the cameras, processes it and analyzes the number of vehicles on the road. The analysis results are used to applications such as controlling the traffic signal lights. Other applications include information for the tourists, and example of which would be delivering real-time video of tourist spots of various locations in the city to the PDAs. In such applications, unlike the traffic flow control system, the video would be delivered directly to the user device through the streaming server. The distribution network manages the QoS for supporting various types of user devices. For example, if the information obtained from the interaction with the user device specifies that QCIF size video will be serviced at 300 kbps, the information is sent to the SLiM server of the camera network to transmit scalable encoded video stream.

For the reasons explained in Section 1.1, SLiM has two-stages. The distribution network is again divided into two stages so that SLiM provides a structure that easily supports multiple users as well as various application servers. Such a structure has an advantage of a single framework being able to support many services. The camera network explained in Figure 4 involves the camera server referred to as the SLiM server being responsible for media encoding in addition to camera array managing. The distribution network contains an internal streaming server that manages multimedia streaming and media QoS.

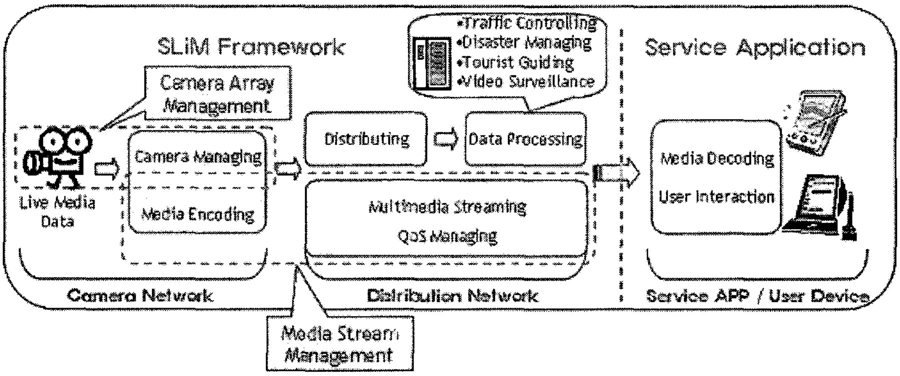


Fig. 4. Conceptual Media Processing Steps in SLiM

2.1. Camera Array Management

Multiple cameras are installed in various locations in a U-City, and a framework is required for methodically managing the cameras. For this purpose, a middleware-level camera management mechanism is introduced for this paper. A camera network shall be defined as a group consisting of multiple camera servers(SLiM servers). A SLiM server has three major functions as exhibited in Figure 5. A SLiM server performs transcoding and scalable encoding in order to compress the multimedia data captured by the cameras, and contains the camera array module for managing multiple cameras in real time. The camera array management module is designed to control multiple cameras from the SLiM server. Since the camera widget state information needs to be shared among the SLiM servers, the data can be read and updated through an external DB server when necessary. The information is organized into a camera widget state table and managed by the list updater of the camera array management module.

The cameras and the SLiM server are in many-to-one relationship. In other words, a single camera transmits video to a single SLiM server, which performs scalable encoding and converts the camera video into various forms of stream. The information is then delivered to the distribution network.

The clear distinction of the media encoding technique is the fact that there are independent and separate modules for calculating the QoS parameter according to the capability and network condition of the user device and for performing media encoding. The former is located in the distribution network, and the latter in the camera network for this framework. This section explains the camera array management technique deployed in the camera network.

Figure 5 displays the internal structure of the camera array management module. Its role is managingthe camera array, the first of the two major areas of study for the proposed framework. The module is a result of modifying parts of the WIF of a previous research project, and it can interact with other components within the middleware.

The detailed description of interaction diagram among modules and protocol is provided in another referred paper [3].

2.2. Media Stream Management

In this research, we designed media stream management module as figure 6 for adaptive streaming to various user devices. As shown in figure 6, the target bitrate information delivered to the media encoder is just used as the category information for determining the level of scalable encoding. For example, if all sessions can be serviced by the base layer encoding, the media encoder does not encode the enhancement layer. In this way, the selective scalable encoding method saves the resources of the complete system.

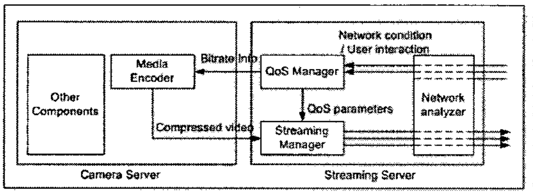


Fig. 6. Media management in SLiM Framework

The media encoder of the camera server gets the target bitrate information in order to use as the encoding option from the QoS manager of the streaming server. The followings addresses the media streaming process

1) User device requests streaming service from user device
2) QoS Manager performs resource management module by using request information and camera information
3) Media Encoder encodes the H.264 scalable layers as resource allocating information
4) Streaming Manager extracts bitstream as capability of user device and information of resource allocating
5) Streaming Manager streams media contents after packetizing

Firstly, the network analyzer informs the QoS manager of the network status between the user device and the streaming server. Next, based on the camera information that the QoS manager obtains from the camera array manager and the resource management model, the QoS manager determines the transmission rate of the user session. Then, this information is transmitted to the camera server and adjusts the frame rate, number of layers and filters of the encoding configuration option of the H.264 scalable extension. After encoding as these adjusted options, the framework extracts bitstream from encoded data according to capability information of user devices and streams it to the user.

The detailed description of our resource management model is provided in referred website [4].

3. Conclusion and Future Works

This paper proposes the camera and media stream management techniques in the middleware level required for implementing the U-City. The study focuses on overcoming the difficulties associated with developing middleware capable of processing and streaming multimedia data from a large number of cameras by expanding the traditional media processing technology. Therefore, this paper addresses the integrated framework which can process and distribute the multimedia stream acquired from the camera array. The SLiM framework which can be used in the various fields efficiently provides the large-scale camera media delivering mechanism to the various user devices. It consists of camera network and distribution network. The camera network is the stage responsible for camera array management as well as encoding the multimedia data provided by the cameras. The distribution network plays the role of efficiently distributing the multimedia data acquired by the camera network. And the user device services variable user requests by using IMS(Interactive Media Framework)[5].

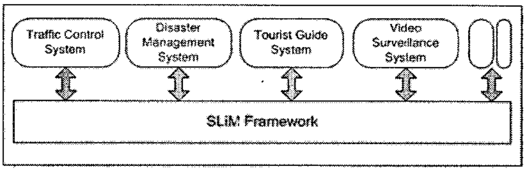


Fig. 7. Several applications of the SLiM framework

The below figure 7 shows the service example using the SLiM framework. The result of this paper can support diverse services such as video surveillance[6], traffic control, disaster evacuation and tourist guiding. Studies of the SLiM framework are currently under way. Firstly, techniques for managing various cameras and supporting user devices are researched. And now, techniques for implementing camera network and efficient scalable encoding are under studying. Afterward, the load balancing technique between the camera servers is planning to be studied. The large scale live media streaming framework developed through this paper will be used in various fields of U-City

References

[1] Manuel Esteve and Carlos E.Palau, "A Flexible Video Streaming System for Urban Traffic Control", Proc IEEE Int'l Conf, IEEE press January-March 2006, pp78-83.

[2] Christopher K. Hess, Manuel Roman, and Roy H. Campbell, "Building Applications for Ubiquitous Computing Environments", In International Conference on Pervasive Computing (Pervasive 2002), pp. 16-29, Zurich, Switzerland, August 26-28, 2002.

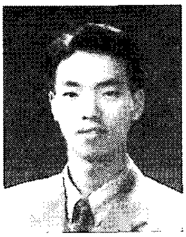
[3] Eun-Seok Ryu, Jung-Seop Hwang, Chuck Yoo, "Widget Integration Framework for Context-aware Middleware", Lecture Notes on Computer Science, Vol. 3744, pp.161-171, Oct. 2005.

[4] WIF, web site: http://os.korea.ac.kr/mediateam/ WIF .htm

[5] Eun-Seok Ryu, Chuck Yoo, "An Approach to Interactive Media System for Mobile Devices", Proceedings of the 12th ACM International Conference on Multimedia (MM 2004), Oct, 2004.

[6] S. Desurmont "A generic flexible and robust approach for intelligent real-time video surveillance systems", Proceedings of the SPIE - Real-time imaging VIII, Vol 5297, No 1, Jan. 2004

◐ 저 자 소 개 ◑



류 은 석 (Eun-Seok Ryu)

1999년 고려대학교 컴퓨터학과 졸업(학사)
2001년 고려대학교 컴퓨터학과 졸업(석사)
2001년 ~ 현재 고려대학교 컴퓨터학과 박사과정(수료)
관심분야 : Multimedia Streaming, Mobile System, Ubiquitous Computing
E-mail : esryu@os.korea.ac.kr

유 혁 (Hyuck Yoo)

1982년 서울대 전자공학과 학사
1984년 서울대 전자공학과 석사
1986년 University of Michigan 전산학 석사
1990년 University of Michigan 전산학 박사
1990년 ~ 1995년 Sun Microsystems Lab. 연구원 & Sun Microsystems Lab. 초빙연구원
1995년 ~ 현재 고려대학교 컴퓨터학과 교수
관심분야 : Embedded System, Multimedia Network
E-mail : hxy@os.korea.ac.kr