

Development of a Scalable Clustering Streaming Server for the Internet Personal-Live Broadcasting

Sangmoon Lee, Sinjun Kang, Byungseok Min, Hagbae Kim

Department of Electrical and Electronic Engineering, Yonsei university, Seoul, Korea

hbkim@yonsei.ac.kr

1. Introduction

As multimedia associated technologies on the internet are rapidly improved, a variety of services to meet sophisticated requirements have increased in the field. There were previous works to improve the QoS(Quality of Service) for the multimedia streaming on the current best-effort internet[1,2] with partial success. The authors of [3] presented principal techniques and algorithms for such streaming video over internet as broadcasting data compression, streaming server, and caching. In particular, with a view to guaranteeing QoS over multimedia server, there were apparent treated resource-management mechanisms embedded in a middle-ware layer[4], an adaptive QoS server model to cope with a variety of circumstances[5], and its implementation[6]. In addition, the authors of [7-8] proposed to achieve the reliable multimedia services that primarily handling continuous audio and video data beyond dealing with the text and static image, in the conventional web services.

In the paper, we developed the internet personal-live broadcasting server system enable to guarantee both High Availability(HA) and scalability. In terms of the service quality, it provides the user interface to create and/or participate a live-broadcasting channel with only basic equipments such as a PC camera and a sound card. It is possible to build multiple channels in a broadcasting server, so that several live broadcastings are concurrently on the air. Also, high scalability is guaranteed to meet dynamic situations where the number of participants in a channel is frequently varying. In the light of system architecture, a cluster of streaming unit can maintain flexible scalability enough to facilitate including or releasing the single unit from the cluster. Furthermore, the CSM(Cluster Server Manager) is responsible for managing the entire broadcasting server system and maintains HA to support continuous service even in the presence of partial failure of a respective streaming unit. It also offers transmission mode switching between TCP and UDP mode by deciding the transmission mode according to network environments and received packet loss of the user system.

2. Internet Live-Broadcasting Server System

2.1 System architecture

The system is composed of one web server, a CSM with backup, and multiple streaming units as depicted in Fig. 1.

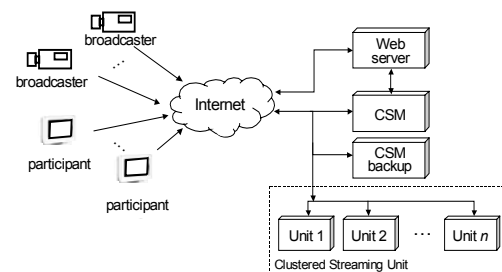


Fig.1 Overall broadcasting system architecture

The web server offers the user application in the form of ActiveX[9] which allows one to create or participate in the live broadcasting. All the broadcasters and participants need accessing to web server to make use of programmed services. It periodically synchronizes all information of the current broadcasting such as broadcaster names, channel titles, total numbers of participants with the CSM. Based on this information, the CSM creates a new channel at the group in the streaming unit.

The CSM plays a key role of controlling broadcasting server system as well as managing individual streaming units. It also balances requests of the participants to join the active channel, manages the system resources through expanding or releasing groups and handles the administrator's command in terms of system monitoring tools. The clustered streaming units that are real broadcasting servers are directly connected with the users while handling actual broadcast contents. For guaranteeing HA, a backup of the CSM uses a Heartbeat[10] in order to detect the CSM failure.

2.2 Channel expansion

Basically one streaming unit consists of fifty groups, and each group can afford ten participants. As the number of participants in a broadcasting group increases, the CSM makes a connection with an empty group in the same streaming unit. But, if it does not have any available one, the CSM attempts to conduct in another unit as depicted in Fig. 2.

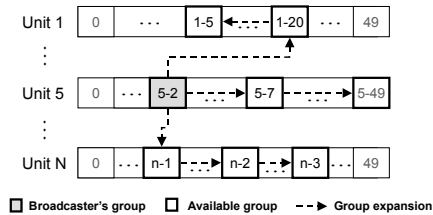


Fig.2 Hierarchical structure of the expanded channel

It periodically updates the information about expanded structure of the group for each channel, and allocates new participants to a proper group according to this information.

2.3 Broadcasting mode switching

Our system supports a flexible transmission mode switching between TCP and UDP in order to transmit broadcasting data efficiently[11]. In normal conditions, the live broadcasting is provided by using UDP. The user system periodically checks data loss or broadcasting status so that it asks the streaming unit to use TCP. The internal buffer in each group has a circular memory structure like Fig. 3, where new data is added and updated continuously and transmitted by data block in the TCP mode.

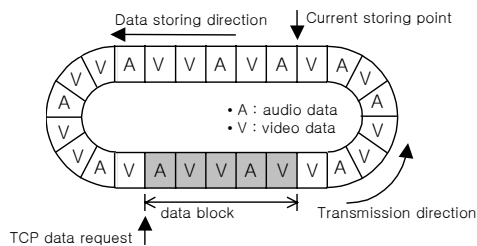


Fig.3 The structure of the buffer in a group for TCP transmission

3. Conclusion

In the paper, we developed an internet live-broadcasting server system. In the aspect of service, it provides the user interface to create and/or participate a live-broadcasting channel only with simple equipments. Multiple channels can be built in a streaming unit, where each group can be expanded as participants increase to meet the reliable services. In the system architecture, a clustered streaming unit supports both flexible scalability and HA to support continuous service in the presence of partial failures in the cluster. In addition, according to broadcasting status it can deliver better services to the users by properly selecting either TCP or UDP mode transmission.

Reference

- [1] B. Furht, R. Westwater, J. Ice, "Multimedia broadcasting over the Internet. I," IEEE Multimedia, vol. 5, Issue 4, pp. 78-82, Oct.-Dec. 1998.
- [2] M. Hofmann, K. Sabnani, "Streaming and broadcasting over the Internet," Proc. of the IEEE Conf. on High Performance Switching and routing, pp. 251-256, 2000.
- [3] W. Dapeng, Y. Hou, Z. Wenwu, Z. Ya-Qin, J. Peha, "Streaming video over the Internet: approaches and directions," IEEE Trans. on Circuits and Systems for Video Technology, vol. 11, Issue 3, pp. 282-300, Mar. 2001.
- [4] T. Abdelzaher, K. Shin, "QoS provisioning with qContracts in web and multimedia servers," Proc. of IEEE Real-Time Systems Symposium, Dec. 1999
- [5] J. Reumann and K. Shin, "Adaptive Quality-of-Service session management for multimedia servers," Proc. of 8th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video, Jul. 1998.
- [6] W. Aref, D. Braun, D. Bushmitch, I. Kamel, S. Mukherjee, "An inexpensive, scalable, and open-architecture media server," Proc. First IEEE/Popov Workshop on Internet Technologies and Services, pp. 36-43, 1999.
- [7] S. Ghandeharizadeh and R. Muntz, "Design and implementation of scalable continuous media servers," Parallel Computing, vol. 24, Issue 1, pp. 91-122, Jan. 1998.
- [8] S. Ghandeharizadeh, R. Zimmermann, W. Shi, "A scalable continuous media server," Multimedia Tools and Applications, Vol. 5, No. 1, pp. 79-108, Jul. 1997.
- [9] <http://www.microsoft.com/com/tech/activex.asp>
- [10] <http://linux-ha.org>
- [11] I. Cidon, R. Rom, A. Gupta, C. Schuba, "Hybrid TCP-UDP transport for web traffic," IEEE Int. Performance, Computing and Communications Conf., pp. 177-184, 1999.