Cross-layer Optimized Multipath Video Streaming over Heterogeneous Wireless Networks

Oh Chan Kwon* and Hwangjun Song†

*Department of Computer Science and Engineering, POSTECH, Pohang, Republic of Korea

E-mail: ochanism@postech.ac.kr Tel: +82-54-279-5684

† Department of Computer Science and Engineering, POSTECH, Pohang, Republic of Korea

E-mail: hwangjun@postech.ac.kr Tel: +82-54-279-2246

Abstract— **In this work, we propose a cross-layer optimized multipath video streaming system over heterogeneous wireless networks. The proposed system uses multiple paths over wireless mobile networks in order to satisfy the quality-of-service requirements for seamless high-quality video streaming services and adopts fountain codes to handle the effect of lost packets over wireless networks efficiently. Moreover, a cross-layer approach is used to determine the code rate of fountain codes efficiently. Finally, the proposed system is implemented and tested in real environments.**

I. INTRODUCTION

The demand for video streaming services over wired and wireless networks has been rapidly increasing. Due to the fast development of communication and networking technologies, video streaming services are available anytime and anywhere. However, it is well known that the resources available over wireless networks are very limited compared to those of wired network. In addition, video streaming services consume significantly more network resources than traditional data services. To reduce the amount of data, the use of effective video compression algorithms is indispensable. Thus, digital video coding techniques have advanced rapidly. International standards such as MPEG-2 [1], MPEG-4 [2], H.263/+/++ [3], and H.264 [4] have been established or are under development to accommodate different needs by ISO/IEC and ITU-T. The compressed video data is generally of variable bit rates due to the generic characteristics of entropy coder and scene change/inconsistent motion change of the underlying video. Furthermore, video data is strictly time-constrained. These facts make the problem of supporting seamless high quality video streaming services over wireless mobile networks more challenging.

Cross-layer designed video streaming technology has been widely studied to provide seamless video streaming services and enhance streaming video quality with minimal redundancy over time-varying wireless networks. By sharing information among layers and/or jointly determining control parameters at multiple layers, cross-layer technology can efficiently deal with the uncertainty inherent in wireless mobile networks. However, it still has some difficulties in adequately supporting seamless high quality video streaming services. In recent years, heterogeneous wireless networks are co-existing to support the variety of subscriber requirements. For example, WLAN (Wireless Local Area Network), standardized by the IEEE 802.11 series, can provide high data throughput while its coverage range is only 50 - 100 m. Cellular-based networks such as GPRS (General Packet Radio Service) and UMTS (Universal Mobile Telecommunications System) can cover a wide area at the cost of decreased data throughput. Lately, WiMAX (Worldwide Interoperability for Microwave Access), HSDPA (High-Speed Downlink Packet Access), CDMA2000 1X EV-DO (Evolution-Data Optimized), and 3GPP LTE (Third Generation Partnership Project Long Term Evolution) have been proposed to accommodate subscriber's requirements. However, it is still challenging to support seamless high-quality multimedia services through only a wireless network.

In this work, we propose a cross-layer optimized multipath video streaming system over heterogeneous wireless networks. The goal of the proposed system is to guarantee QoS required for a video streaming service. The rest of this paper is organized as follows. In section II, the review of fountain codes is presented. The proposed video streaming system is introduced in section III. Experiment results are provided in section IV. Finally, concluding remarks are given in section V.

II. REVIEW OF FOUNTAIN CODES

Fountain codes [5] such as Luby transform (LT) [6], Raptor [7], and Online [8] codes are block-based forward-error-correction (FEC) scheme that provide non-systematic coding, flexibility, coding efficiency, and ratelessness (i.e., they can generate encoded symbols endlessly from a given finite number of source symbols). These characteristics are very useful for transmitting delay-sensitive data over error-prone wireless networks. Figure 2 shows a simple example of fountain codes. The data stream is divided into source blocks, each of which is composed of K_{source} source symbols. The source symbols are randomly XOR-ed to generate the encoding symbols. The encoding symbols are generated by

$$E_i = \sum_{j \in \mathbf{\Gamma}_i} \oplus S_j,\ i = 1,...,l \text{ and } 1 \le j \le K_{source},$$

where $\mathbf{\Gamma}_i$ is the set of source symbols of the i_{th} symbol, $\oplus$ is the XOR operator, S_j is the j_{th} source symbol, and l is the length of the encoded symbols. The encoding symbols are transmitted to the receiver through an error-prone wireless channel as shown in Figure 2. The decoding process is equivalent to solving linear equations. In general, the message passing [5] decoding algorithm is widely used because of its low complexity. First, the fountain decoder finds an encoded symbol consisting of only a source symbol. This source symbol is directly recovered by the corresponding encoding symbol and is XOR-ed to all the encoding symbols that are mapped to the reconstructed source symbol. The receiver can easily reconstruct source symbols from encoded symbols if the sender-peers and receiver-peer share the seed number and random generator. Hence, the required side information for fountain decoding is negligible. The receiver can successfully reconstruct all source symbols if a sufficient number of encoded symbols are available even when some encoded symbols are lost as shown in Figure 2. In general, the number of encoded symbols required for successful fountain decoding is calculated as

$$K_{\min} = (1+\gamma)\cdot K_{source},$$

where γ is the symbol overhead with a very small real number. This equation indicates that the number of received encoded symbols must be slightly larger than K_{source} to reconstruct source symbols successfully. In the FEC scheme including fountain codes, the code rate (c) plays an important role because it determines the amount of redundant data for error protection; that is, $c = n/K_{source}$, where n is the number of encoding symbols. Accordingly, the control parameters of the fountain encoding process are K_{source} and c . Both of them have to be determined before the fountain encoding operation takes place.

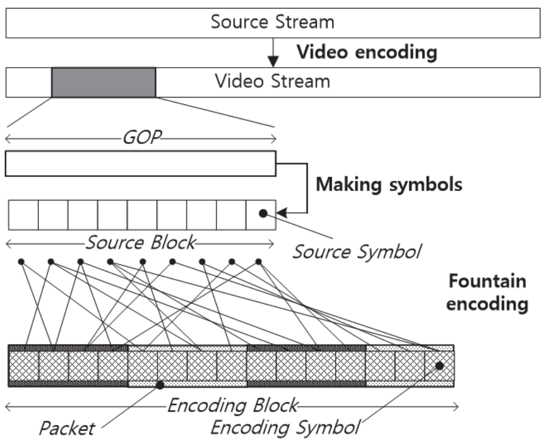

Fig. 1 An example of fountain codes

III. Proposed Video Streaming System

The proposed system is designed to provide a seamless video streaming service using multiple paths. Now, we first explain the architecture of the proposed multipath video streaming system and then determining process of control variables of the proposed algorithm. The proposed system is presented in Figure 2. It consists of rate controller, fountain encoder, parameter control unit, and weighted round robin distributor. In this work, in terms of application layer, we call multiple paths a virtual path.

A. System Architecture

The proposed system includes fountain encoder and weighted round robin distributor. To make fountain encoder and distributor work well, we have to find the code rate for fountain encoder and the transmission rate of each path. Fountain encoding process can be characterized using three factors: source block size, symbol size and code rate. Source block is the basic unit of fountain encoding. The original data have to be divided into source blocks to apply fountain encoding. Source block size can vary. After that, according to symbol size each source block divided into source symbols. Now, we consider how to divide video stream into fountain source blocks. First of all, video stream is divided into GOPs (group of picture) for better error protection because one block loss can affect one more GOPs if the source block is not synchronized with the GOP. Finally, encoded blocks are split into fixed-size packets before transmission. Hence a packet may include several symbols.

As shown in Figure 2, a client must provide its path status information including available bandwidth, delay, and PLR (packet loss rate) to video server. Hence, a client estimates the bandwidth based on SNR (signal-to-noise ratio) and easily calculates the delay using round-trip time. PLR is another factor to characterize the path. In the proposed system, PLR is estimated by RSS (Received Signal Strength) [9]. Instead of PLR of each path, that of a virtual path is more important in terms of video encoder. Now, we define the feedback information as follows:

$$\overrightarrow{bw} = (bw_1, bw_2, ..., bw_m),\ \vec{d} = (d_1, d_2, ..., d_m) \text{ and}$$
$$p \text{ is the packet loss rate of virtual path}$$

where bw_i and d_i are the bandwidth (bps, bits/s) and delay (seconds) of the i_{th} path respectively and m is the number of physical paths consisting of a virtual path. The above feedback information are sent periodically (when the pre-determined time is expired) or when the condition of virtual path changes significantly (i.e. the difference between the current PLR (p) and the previously reported PLR ($p^{previous}$) is larger than the pre-determined threshold as follows.

$$\left| p^{previous} - p \right| > p_{th},$$

where p_{th} is a threshold. Video encoder reports the required maximum delay and BLR (block loss rate) to the parameter control unit before actual video data transmission. Then, the parameter control unit sends back the available encoding rate information after analyzing the established virtual path. Video encoder generates compressed video data according to the received encoding rate.

B. Determination of Video Rate and Code Rate

Now, we can formulate the problem as follows.

Problem Formulation: Find the maximum encoding rate ($VR_{\max}$) and code rate (c) for video streaming

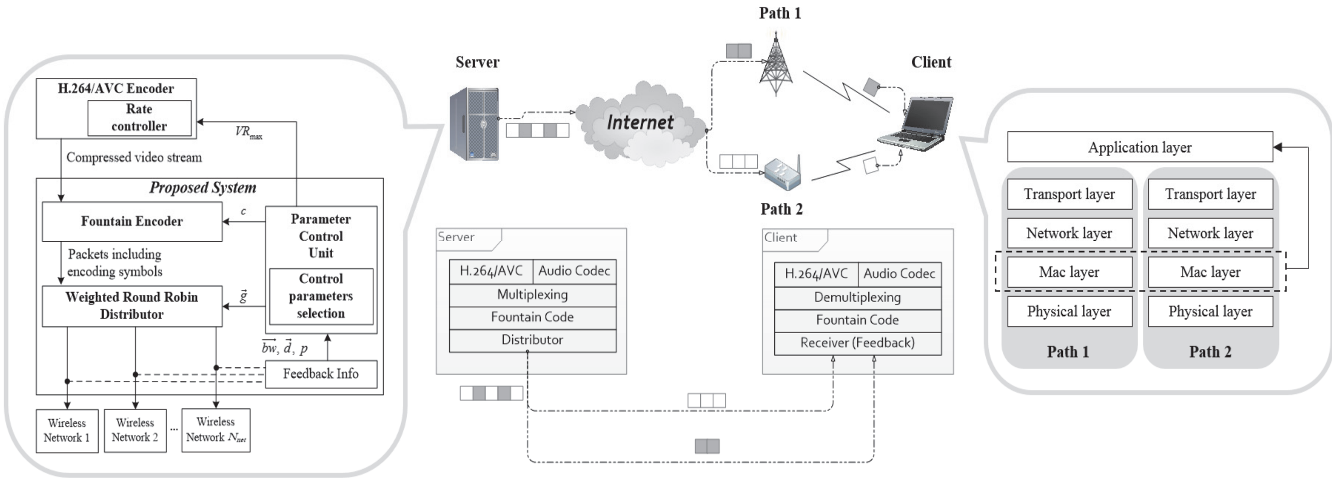

Fig. 2 Proposed system architecture

Subject to $d \le d_{\max}$, and $p_{blr} \le p_{\max}$,

where d is the delay, $d_{\max}$ is the tolerable maximum delay, p_{blr} is the BLR, and $p_{\max}$ is the tolerable maximum BLR. The optimal solution is obtained by the following three steps.

Step 1: Initialization

Since a client sends PLR information of the error-prone virtual path, the proposed virtual path component must calculate BLR based on the received PLR to determine the code rate of fountain encoder. Now we define interim variable K as follows.

$$K = \left\lceil \frac{B_{gop}}{PS} \right\rceil,$$

where $\lceil * \rceil$ denotes the smallest integer that is larger than *, PS is packet's payload size and B_{gop} is the amount of output bits for a GOP.

Step 2: Find the maximum transmission rate ($TR_{\max}$)

The fountain decoder obtains the compressed video data successfully only when the number of arrived packets is larger than that of decodable packets. Now, we define the delay to transmit a packet though the i_{th} path by

$$TD_i = \frac{PS}{bw_i}.$$

Then, the number of packets that are transmitted though the i_{th} path satisfying the delay constraint of the above problem formulation is calculated by

$$g_i = \frac{d_{\max} - d_i}{TD_i}.$$

Therefore, the maximum bandwidth of a virtual path is

$$TR_{\max} = \frac{PS \cdot \sum_{i=1}^{m} g_i}{n_{gop} / fr},$$

where n_{gop} is the number of frames in a GOP and fr is the frame rate of video.

Step 3: Determine the code rate (c)

First the PLR of virtual path can be calculated by

$$p = \sum_{i=1}^{m} g_i \cdot plr_i,$$

where plr_i denotes PLR of the i_{th} path that calculated by RSS [9]. Now we can calculate BLR function $P(K,c)$ as follows.

$$P(K,c) = \sum_{i=t-K'+1}^{t} \binom{t}{i} p^i (1-p)^{t-i},$$

where $t = \lceil K/c \rceil$ and

$$K' = \lceil (1+\gamma) K \rceil,$$

where γ is the proportion of additional symbols for successful fountain decoding. Generally, fountain code has very low value of γ. In this phase, we have to find largest c which satisfy below equation:

$$P(K,c) \le P_{\max}.$$

Consequently, the maximum encoding rate () is determined by

$$VR_{\max} = TR_{\max} \times c.$$

IV. Experiment Results

Java is used to implement the proposed system. During the experiment, we assume that there are only two heterogeneous paths between a video server and a client, that is, 802.11b and 802.11g are simply used as access technology for each path. The tolerable maximum delay and BLR are set to 500 ms and 0.01, respectively. The experiment is performed for 40 seconds. And the reference software of H.264/AVC, JM12.4 [10] and original JM rate control mechanism are used. CIF format HARBOUR video sequence is adopted as a test sequence. The GOP structure is IPPP… and the number of

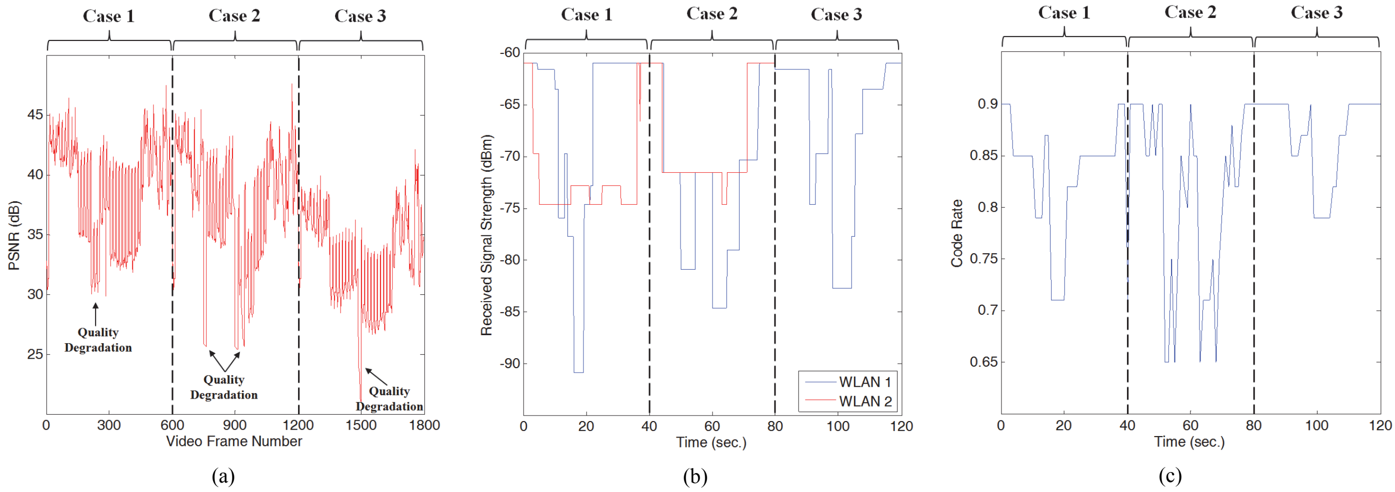

Fig. 3 Plots of experiment results for three cases: (a) PSNR, (b) wireless channel state, and (c) code rate of fountain codes.

frames in a GOP is set to 15. "Foreman", "Soccer", and "Harbour" of CIF size video were employed as test video sequences. We test the following three cases for the performance comparison. ***(Case 1) Proposed system***: The proposed system uses two wireless networks and cross-layer approach for calculating feedback information. ***(Case 2) Virtual path without cross-layer design***: This system also uses two wireless networks but does not use cross-layer approach for calculating feedback information. ***(Case 3) Cross-layer design using single path***: This system is equal to the proposed system, except for using only one wireless network.

The experiment results are presented in Figure 3. Figure 3 (a) shows the resulting PSNR comparison among the three cases. It is obviously observed that Case 1 shows the best PSNR compared to the others. In Case 2, PSNR is degraded when sudden wireless channel quality degradation occurs as shown in Figure 3 (b), because the code rate is not appropriately adjusted as shown in Figure 3 (c). In Case 3, the network bandwidth is insufficient for video streaming since only a single network is used, and thus the media streaming server provides a low-quality video streaming service. The average PSNR and average video bitrates are summarized in Table I.

TABLE I
AVERAGE VIDEO BITRATES AND CORRESPONDING PSNR VALUES.

Algorithm	Average PSNR (dB)	Average video bitrates (Kbps)
Proposed system	38.406	312.323
Virtual path without cross layer design	37.085	292.501
Cross-layer design using single path	32.869	186.475

V. CONCLUSIONS

In this work, we have proposed a cross-layer optimized multipath video streaming system over heterogeneous networks and implemented the whole system by Java. By the experiment in the real wireless environment, we show that virtual path using fountain code consolidates multiple paths for video streaming. This approach support higher bandwidth, lower delay, and lower BLR than traditional approach which use only one network at a time. In addition, seamless video streaming service is possible without a vertical handoff time.

ACKNOWLEDGMENT

This research was one of KOREN projects supported by National Information Society Agency (13-951-00-001) and the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

REFERENCES

[1] ISO/IEC 13818 (MPEG-2), “Generic coding of moving pictures and associated audio information,” November 1994.
[2] ISO/IEC JTC 1/SC 29/WG 11 N4030, “Overview of the MPEG-4 standard,” March 2001.
[3] ITU-T Recommendation H.263 version 2, “Video coding for low bit-rate communication,” February 1998.
[4] ITU-T Recommendation H.264 version 3, “Advanced video coding for generic audiovisual services,” November 2007.
[5] J. C. MacKay, “Fountain codes,” IEE Proceedings Communications, vol.152, no.6, pp.1062–1068, 2005.
[6] M. Luby, “LT codes,” Proceedings of the Annual Symposium on Foundations of Computer Science, pp. 271–280, 2002.
[7] A. Shokrollahi, “Raptor codes,” IEEE Transactions on Information Theory, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.
[8] P. Maymounkov, “Online codes,” NYU, Technical Report TR2002-833, Nov. 2002.
[9] Gyeongcheol Lee and Hwangjun Song, “Cross layer optimized video streaming based on ieee 802.11 multi-rate over multi-hop mobile ad hoc networks,” Mobile Networks and Applications (MONET), vol. 15, pp. 652–663, October 2010.
[10] Joint Video Team Reference Software, JM12.4, http://iphome.hhi.de/suehring/tml/download/, Oct 2007.