Copyright ⓒ 2010 KSII

Encryption Scheme for MPEG-4 Media Transmission Exploiting Frame Dropping

Dongkyoo Shin,[1] Dongil Shin,[1] Jaewan Shin,[1] Soohan Kim[2] and Seungdong Kim[3]
[1] Department of Computer Engineering, Sejong University, 98 Kunja-Dong, Kwangjin-Gu,
Seoul 143-7477, South Korea
[e-mail: {shindk, dshin}@sejong.ac.kr, shinnom@gce.sejong.ac.kr]
[2] Internet Infra R&D Planning Team, NW. Div., Samsung Electronics Co., LTD
416 Metan-3 Dong, Yeongtong-Gu, Suwon, South Korea
[e-mail: ksoohan@samsung.com]
[3] Coporate Reseach Center, SK Telecom Co., LTD
SKT-Tower, 11, Euljiro 2-ga, Jung-gu Seoul 100-999, South Korea
[e-mail: harry220@gmail.com]
*Corresponding author: Dongkyoo Shin

Received April 23, 2010; revised July 5, 2010; revised August 3, 2010; accepted August 11, 2010; published October 30, 2010

Abstract

Depending on network conditions, a communication network could be overloaded when media are transmitted. Research has been carried out to lessen network overloading, such as by filtering, load distribution, frame dropping, and other methods. Among these methods, one of the most effective is frame dropping, which reduces specified video frames for bandwidth diminution. In frame dropping, B-frames are dropped and then I- and P-frames are dropped, based on the dependency among the frames. This paper proposes a scheme for protecting copyrights by encryption, when frame dropping is applied to reduce the bandwidth of media based on the MPEG-4 file format. We designed two kinds of frame dropping: the first stores and then sends the dropped files and the other drops frames in real time when transmitting. We designed three kinds of encryption methods using the DES algorithm to encrypt MPEG-4 data: macro block encryption in I-VOP, macro block and motion vector encryption in P-VOP, and macro block and motion vector encryption in I-, P-VOP. Based on these three methods, we implemented a digital rights management solution for MPEG-4 data streaming. We compared the results of dropping, encryption, decryption, and the quality of the video sequences to select an optimal method, and found that there was no noticeable difference between the video sequences recovered after frame dropping and the ones recovered without frame dropping. The best performance in the encryption and decryption of frames was obtained when we applied the macro block and motion vector encryption in I-, P-VOP.

Keywords: MPEG (Moving Picture Experts Group)-4, frame dropping, encryption

This study was supported by grants from the Korea Health 21 R&D Project, the Ministry for Health, Welfare and Family Affairs. (A040163), the Korean government.

DOI: 10.3837/tiis.2010.10.013

1. Introduction

Because of the continuing development of the Internet and the wireless communication environment, users are becoming interested in a variety of services that demand high quality. To satisfy these users, high quality media services have been provided for the distribution of digital contents. In particular, because of the appearance of multimedia services, there is a demand for broad bandwidth services, such as image transmission, and real-time services, like those provided by broadcasting. In these services, a streaming service using MPEG-4 is a good solution for both wired and wireless service. MPEG-4, which was developed by MPEG (Moving Picture Experts Group), consists of an encoding standard that includes compression encoding for video, audio, static images, computer graphics signals, composition audio, and text. To provide quality services, the continuous stream should be offered by controlling the bit rate. We need a method that is able to control the bit rate of the video based on the state of the network to lessen network overload and realize an effective video streaming service.

Many methods have been studied to reduce network overload, such as filtering, distribution methods, and congestion algorithms. However, a distribution method that stores data in several servers requires large storage away from a centralized environment [1]. Out of the congestion control algorithms, UDP congestion control uses the TFRC (TCP-Friendly Rate Control) technique because TCP congestion control is not an appropriate method for an uninterrupted multimedia streaming service [2]. Although TFRC is appropriate under normal conditions, there could be a large number of packet losses in unpredictable traffic circumstances to be encountered with a streaming service [3].

In this paper, we apply frame dropping, which is able to control the media bandwidth by dropping the B-frames. B-frames have the lowest dependency among the frames. Media Intellectual Property Rights (IPR) have become increasing important as streaming services increase. For IPR protection, a Digital Rights Management (DRM) system restricts the use of digital contents to protect the interests of copyright holders. A DRM system controls the number of views, length of views, altering, sharing, copying, printing, and saving for digital contents. The MPEG on ISO, which standardizes and manages DRM methods, additionally established an Intellectual Property Management and Protection (IPMP) interface standard for a copyright management service [4][5]. In this paper, we apply the DES algorithm for Intellectual Property Management and Protection (IPMP). DES is a block encryption algorithm, which takes 64-bit data as an input and outputs 64-bit encrypted data.

2. Related Work

To transmit contents in a communication network, we need a high bandwidth, but bandwidth changes greatly. We should use a stable and low bandwidth to prevent the changes in video quality that result from variations in bandwidth. MPEG-4 is used to support video data with a low bit rate because it has remarkable compression efficiency, better than H.261, MPEG-1, and MPEG-2.

Hence, MPEG-4 techniques are of great use in transmitting video in a mobile communication network and internet broadcasting system [6][7]. We illustrate the traditional frame dropping and encryption algorithms in the following sections.

2.1 Traditional MPEG Video Dropping Algorithm

The frame dropping that is proposed in this paper drops specific video frames to lessen the server overload. We drop B-frames and then drop I-and P-frames according to the dependency among the frames. In addition to frame dropping, various methods exist to lessen the server overload.

The Heidelberg transport system (HeiTS) works in real-time when transmitting. A major issue with HeiTS is its responsiveness, that is, how it adapts to the available bandwidth. HeiTS uses scaling up and scaling down according to the server overload [8].

A dropping delivery algorithm by block dropping transmits compressed video data by reducing the bit rate in the server when the server bandwidth is reduced. At this time, the method that omits the block by the I-, P-, and B-frames is the block dropping algorithm. In this paper, a block is a macroblock (MB) [9].

Abstraction for continuous media skips or purges a portion of the frames when transmitting breaks under a time restriction [8][10].

Coefficient dropping removes the VLC code of a high frequency DCT coefficient that is not important. Coefficient dropping can control a small amount more closely than frame dropping. There are two methods. One method removes sequences in EOB (End of Block) code. The other method removes a coefficient that is not important apart from the location in a single block [11].

Scalable video coding (SVC) transmits a scalable bit stream for the receipt of various quality video. SVC has an advantage due to its capability of reconstructing lower resolution or lower quality signals from partial bit streams. SVC has been used to enable the encoding of a high quality video bit stream that contains subset bit streams. Bit streams based on SVC can be decoded with similar complexity and reconstruction quality to the bit streams achieved using the MPEG4 AVC design for the same quantity of data as in the subset bit stream [12][13][14].

2.2 MPEG Video Encryption Technique

A naïve algorithm was proposed by I. Agi and L. Gong. This naïve algorithm is used to encrypt entire MPEG bit streams by a standard encryption algorithm like DES. It deals with the MPEG bit streams as plain text and takes no advantage of the structural merits of MPEG [15].

A selective algorithm was proposed by T. B Mayples and G. A. Spanos. Algorithms that exploit the features of the layered structures of MPEG are classified as Selective Algorithms [16].

Tang proposed the ZigZag-permutation algorithm. ZigZag algorithm encryption incorporates an integral part of the MPEG compression process in the ZigZag-Permutation Algorithm. A random permutation list to map the individual 8 x 8 blocks to a 1 x 64 vector is replaced with mapping the 8 x 8 blocks to a 1 x 64 vector in ZigZag order [17].

A video encryption algorithm was proposed by L. Qiao and Nahrstedt. This algorithm is a symmetric cryptosystem utilizing the statistical analysis of compressed video frames and properties of MPEG. This algorithm encrypts all of the data in the MPEG picture layer using a DES encryption algorithm [18].

Efficient and fully scalable encryption is processed by FGS (Fine Granularity Scalability) middle stages directly on the cipher text without decryption. This algorithm was proposed by Chun Yuan and Yuzhuo Zhong [19].

A joint signal processing and cryptographic approach to multimedia encryption proposes two atomic encryption operations that can preserve standard compliance and are friendly to delegate processing [20].

Selective video encryptions based on AVC (advanced video coding), Partial Encryption of Intra-prediction Mode (PEM), Partial Encryption of Coefficients (PEC), Partial Encryption of Motion Vectors (PEV) and Key Generator also have been proposed. The encryption processes can be combined with the compression process. Such an algorithm was proposed by Shiguo Lian [21].

Partial encryption was proposed by Amir Said. Only a fraction of the data bits are encrypted to reduce computational complexity. Customized versions are provided to different users according to their permission [22].

3. Design and Realization of Frame Dropping

In this paper, we designed an MPEG-4 frame dropping technique. Frame dropping drops B-frames to maintain video quality while lowering the bit rate based on the dependency among the frames. The frame dropping structure is shown in **Fig. 1**.

The first dropping method drops frames before the transmission of MPEG-4 media files and then sends the dropped files to the client. The second method drops frames in real time while transmitting.

(1) Store in server and send the dropped files *(dropping-1)*
(2) Dynamically drop frames in real-time when transmitting *(dropping-2)*

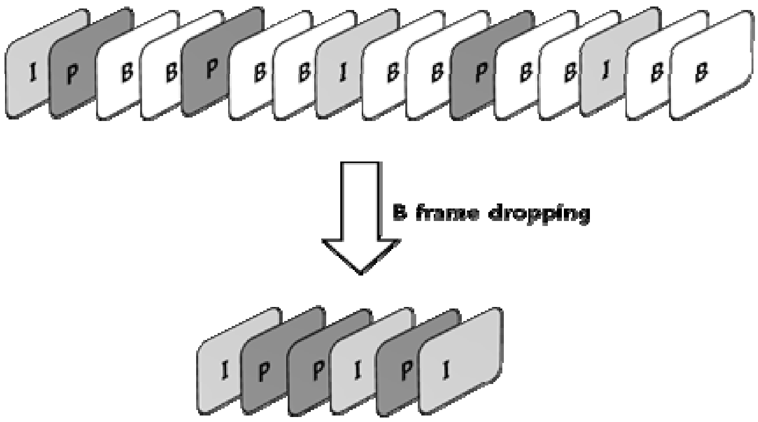

Fig. 1. B-frame dropping.

File format modification is not inevitable in the *dropping-1* method. The MPEG-4 file format is based on the apple computer QuickTime file format developed [23] and then adopted in the ISO media file format part 12 [24].

An MPEG-4 file can be divided into data and metadata. The metadata includes information about how the data is stored and technical description [25]. **Fig. 2** shows the structure of the MPEG-4 file format. At the highest level, the movie atom (MOOV) contains other types of atoms. The leaf atoms of the MOOV atom contain detailed information from the actual data.

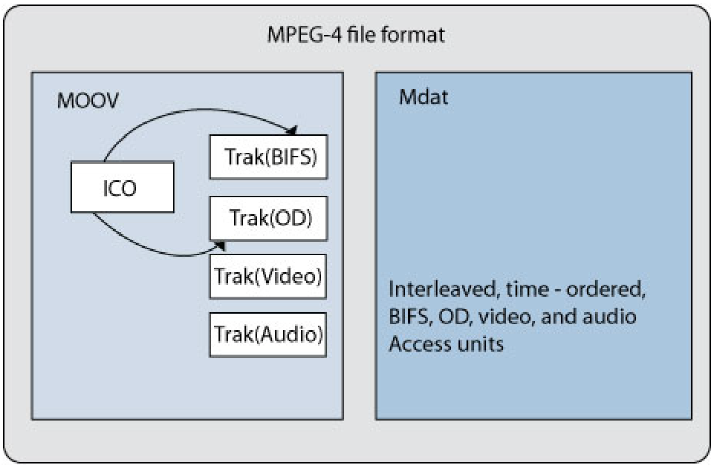

Fig. 2. MPEG-4 file format.

3.1 Storing in the Server and Sending the Dropped Files *(dropping-1)*

This algorithm stores the B-frames dropped video data files in the server and then sends the files to the client. To implement frame dropping, B-frames are dropped, I- and P- frames are moved to the locations of B-frames, and then the related atoms should be modified since the atoms are associated with each other. The Stsz atom is the sample size, Stco atom is chunk offset, Mdat atom is frame media data, and Stts atom includes the time to sample.

When B-frames are dropped, modifications of each atom are performed by following steps.

(1) Stsz atom information is modified to fit a new frame size.
(2) Stco atom information is modified to correct the starting point of the next frame.
(3) Any unnecessary media data in the Mdat atom is deleted because Mdat atom has the previous media data.
(4) The play time for sample information in the Stts atom is modified to reflect the play time of the previous deleted frames.

Fig. 3 shows the frame dropping result of PrettyWoman.mp4. The first figure in **Fig. 3** is the original data, the second figure is the result after B-frame dropping, the third figure is the result after B-frame dropping and Stsz atom modification, and the last figure is the result after B-frame dropping and Stsz and Stco atom modification. **Fig. 4** shows the result from the frame dropping of Akiyo.mp4.

Since each VOP (Video Object Plane) has a distinct video-start-code based on MPEG-4 video standards [23], frames are extracted from the video data in Mdat, B-frame is dropped and I- and P-frames are moved to the locations of B-frames. To access each frame, the information about the start point of the frame is needed and it can be obtained from Stco and Stsc atoms. To proceed with frames, the size of each frame is needed and it is contained in the sample size of the Stsz atom.

After B-frame dropping, video sequences with partially destroyed frames are obtained as show in the second figure in **Fig. 3** and **Fig. 4**, respectively, since the Stsz atom keeps the size of the dropped B-frames and I- and P-frames are moved to locations of the dropped B-frames.

After the Stsz atom is modified, video sequences with partially destoryed frames are obtained again, as shown in the third figure in **Fig. 3** and **Fig. 4**, respectively, because the Stco atom is not fixed. The Stco atom is the chunk offset and the chunk size is changed as the sample size is changed.

After the Stco atom is modified, video sequences without any error are obtained, as shown in last figure in **Fig. 3** and **Fig. 4**, respectively.

Fig. 3. Pretty Woman dropping result.

Fig. 4. Akiyo dropping result.

Then, any unnecessary media data after the last frame in the Mdat atom is deleted and the play time information in the Stts atom is modified.

3.2 Dynamically Dropping Frames in Real-Time when Transmitting *(dropping-2)*

This algorithm drops B-frames in real-time transmitting when sending the media file. This algorithm exploits the MPEG-4 format and header structure and sends the frames without B-frames. I-VOP, P-VOP, and B-VOP are discerned by the VOP_coding_type. This method plays the I- and P-frames at the B-frame playtime in the original media. This method can drop the frame without header modification.

4. Encryption in Frame Dropping

We designed and implemented a DRM solution for a dropped MPEG-4 media file for safe and fast transmission in a ubiquitous environment. The DRM system structure is shown in **Fig. 5**.

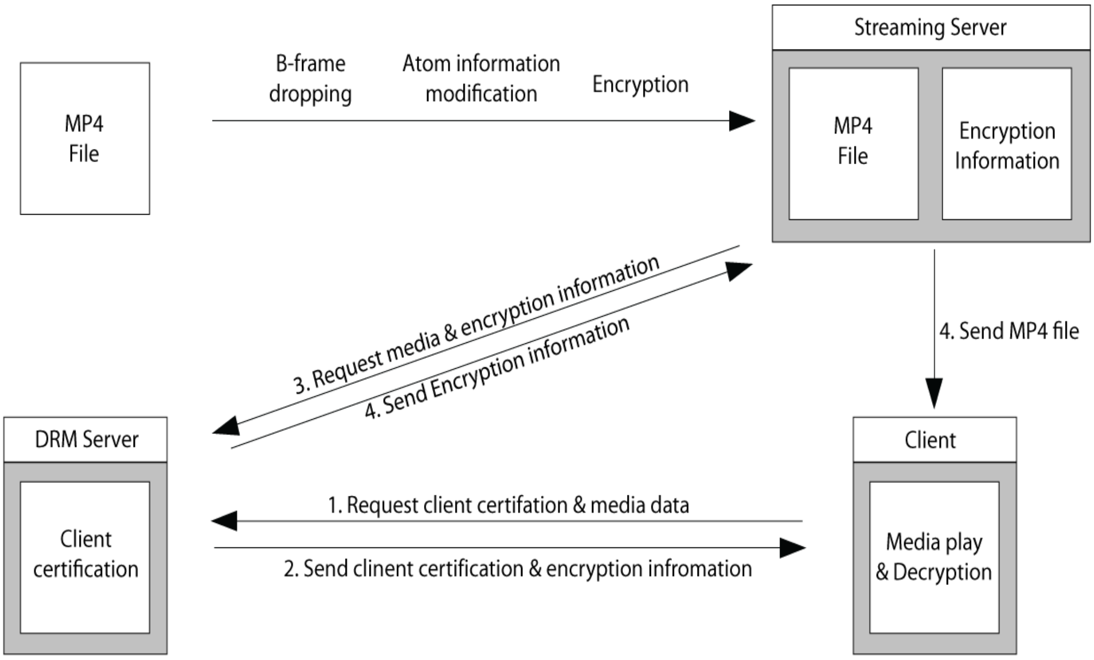

Fig. 5. DRM system structure on streaming service.

The encryption method exploits the MPEG-4 format and header structure, and encrypts the MB and MV frames using selective encryption. The DES encryption algorithm is used for MB and MV encryption.

Each example of VOP has a video_start_code [23]. The Video_start_code value is 00, 01, B6 in hexadecimal. I-VOP, P-VOP, and B-VOP are discerned by the Vop_coding_type [23][26].

We applied the three kinds of encryption methods to the video data:

1) Encryption of macro blocks (MBs) in I-VOP *(encryption-1)*
2) Encryption of MBs and motion vectors (MVs) in P-VOP *(encryption-2)*
3) Combination of Encrypting MBs and MVs in I-VOP and P-VOP *(encryption-3)*

4.1 Macro Block Extraction from I-VOP and Its Encryption (*encryption-1*)

After I-VOP is extracted from the video data, the DES algorithm is applied to the I-VOP. The DES algorithm is a block algorithm, which takes 64-bit data as an input and outputs 64-bit encrypted data, without affecting the size of the data. If we use an encryption algorithm, which has different input and output sizes, it should be changed because the offset, size, and structure of the MPEG-4 file are affected by the algorithm. We chose the DES for this reason. We only applied multiples of 64 as an input to avoid changing the file size and prohibit DES padding from occurring. The second image in **Fig. 6** shows the result of the encryption of MBs in I-VOP.

Fig. 6. DRM encryption result of dropped media.

4.2 Encryption of Macro Block and Motion Vectors in P-VOP (encryption-2)

After P-VOP is extracted by exploiting the MPEG-4 file structure and format, intra MBs and MVs are encrypted as I-VOP encryption. The third image in **Fig. 6** shows the results of encrypting MVs and MBs in P-VOP.

4.3 Encryption of Macro Blocks and Motion Vectors in I-VOP and P-VOP (encryption-3)

Each encryption method alone does not provide satisfactory results because the intra-coded MVs preserve a lot of coding information. To solve this problem, we encrypted all of the MBs and MVs in the I-VOP and P-VOP. The last image in **Fig. 6** shows the encryption of the MBs and MVs. This would provide the strongest encryption effect, compared to applying each method respectively, but the amount of data for encryption is increased, which decreases the performance at the client when playing back the streaming sequence.

5. Simulation on the Speed of Frame Dropping and Data Encryption

In this paper, we proposed two dropping methods and three encryption methods for MPEG-4 media. For the simulation, we modified and reprogrammed source code of the MPEG4IP [27], which is a tool for streaming video and audio that is standards-oriented and free from proprietary protocols and extensions. In the simulation, we used the movie files: PrettyWoman.mp4 and GoneWithTheWind.mp4. Also, we used the MPEG-4 sample media files: Akiyo.mp4, Foreman.mp4, Hall_monitor.mp4, Coastguard.mp4, Container.mp4, News.mp4, and Stefan.mp4. **Table 1** and **Table 2** give summaries of the samples.

Table 1. The characteristics of the MPEG-4 movie files

Data file name		PrettyWoman.mp4	Gone with the wind.mp4
File Size		172MB	240MB
# of samples	Total	89916	124738
	I-VOP	10403	14432
	B-VOP	71932	99790
	P-VOP	7581	10516
Bitrates (Kbps)		400Kps	3475Kbps
Duration (Sec)		1:02:34	50:38
Frame rates (fps)		89916	23000
Width x Height		736 x 384	640 x 464

Table 2. The characteristics of the MPEG-4 sample files

Data file name		Akiyo .mp4	Foreman .mp4	Hall_monitor .mp4	Coastguard .mp4	Container .mp4	News .mp4	Stefan .mp4
File Size		111	4145	3426	528	2336	207	798
# of samples	Total	100	400	300	200	500	100	200
	I-VOP	9	35	26	18	44	9	18
	I-VOP	66	265	199	132	331	66	132
	I-VOP	25	100	75	50	125	25	50
Bitrates (Kbps)		222	2826	2336	1075	1908	418	1629
Duration (Sec)		4	12	12	4	10	4	4
Frame rates (fps)		25	25	25	25	29.970	25	25
Width x Height		176 x 144	352 x 288	352 x 288	176 x 144	352 x 288	176 x 144	176 x 144

5.1 Simulation Results on Frame Dropping

In this paper, we proposed two dropping methods. *Dropping-1* has the advantage of server optimization by saving the dropped files; the advantage of *Dropping-2* is that the frames are dropped without changing any existing media header structures or the original data. In the *Dropping-1* method, the media file size can be reduced. **Fig. 7** shows the file size changes after applying *Dropping-1*. The media file size is reduced to 58% of the original media file by dropping B-frames.

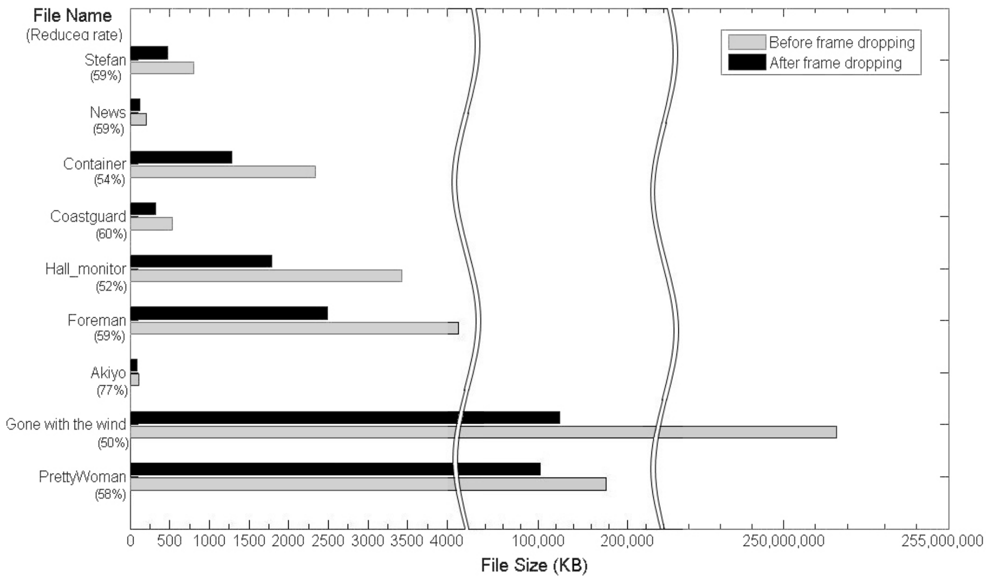

Fig. 7. File size change after applying *Dropping-1*.

There is little difference between the media data with B-frames dropping and the original media data. There also is negligible required time to modify the atoms. **Fig. 8** shows the time for modifying atoms in the media.

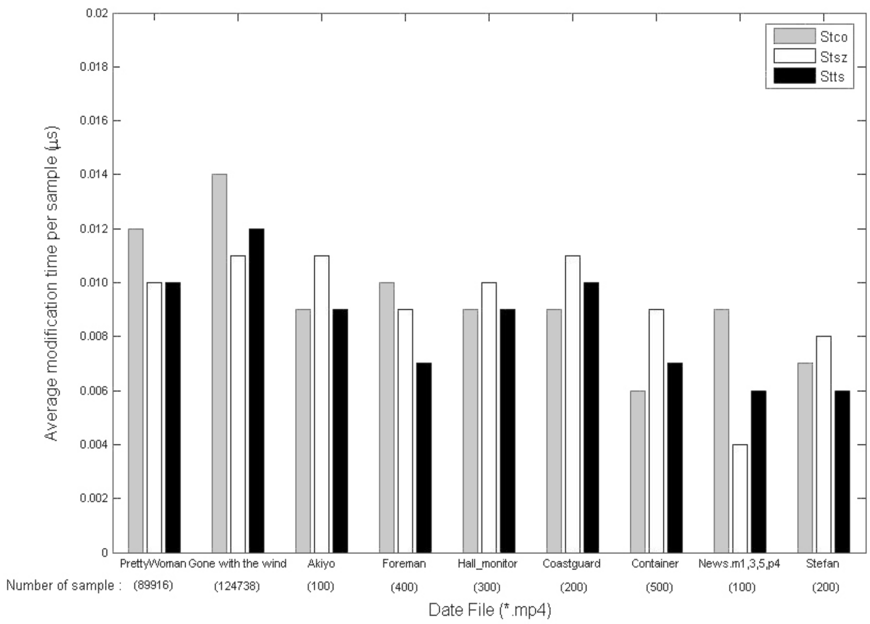

Fig. 8. Time for modifying atoms.

Dropping-2 only takes the replay time of the I- and P- frames received, replacing the dropped B-frames without modifying the header structures and media data. **Fig. 9** shows the replay time from real-time frame dropping, that is *Dropping-2*.

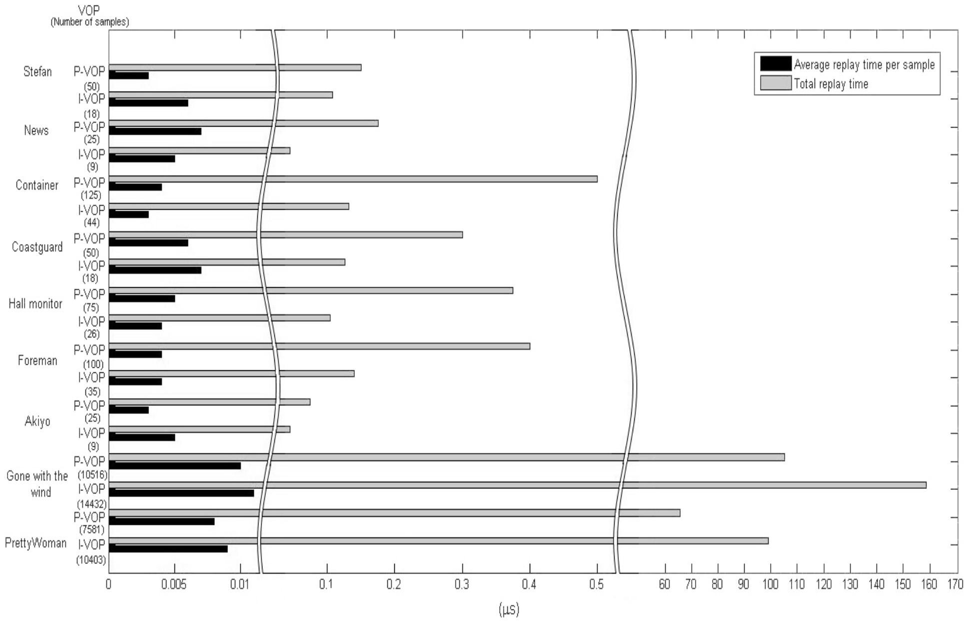

Fig. 9. Replay time from real-time frame dropping **(***Dropping-2*).

Both dropping methods can lessen network overload when sending a B-frames dropped media file and consume almost the same time in dropping. Each method can be used to play high quality media.

5.2 Simulation Results on Video Date Encryption

In this research, we proposed three different encryption methods for MPEG-4 media data. While *encryption-3* offers the strongest encryption results among the three methods, it requires a large amount of data for encryption and takes time to process. Even though we can control the amount of data for encryption by adjusting the I-VOP ratio, to encrypt only I-VOP macroblocks, in general, requires small amounts of data for encryption. The time for encrypting VOP is represented as follows:

$$E(t) = DES(t) + M(t). \tag{1}$$

$E(t)$ is the processing time for the encryption of VOP, $DES(t)$ is the time for DES algorithm processing, and $M(t)$ is the processing time for pre-processing.

Even though *encryption-3* requires time for encryption and decryption that is a multiple of the time needed for *encryption-1*, this may not affect the playback, because the client processes decryption, decoding, and rendering in its memory or pre-process buffer of the swap memory area. **Fig 10** gives a summary of the encryption time from 3 encryption methods after applying the frame dropping method.

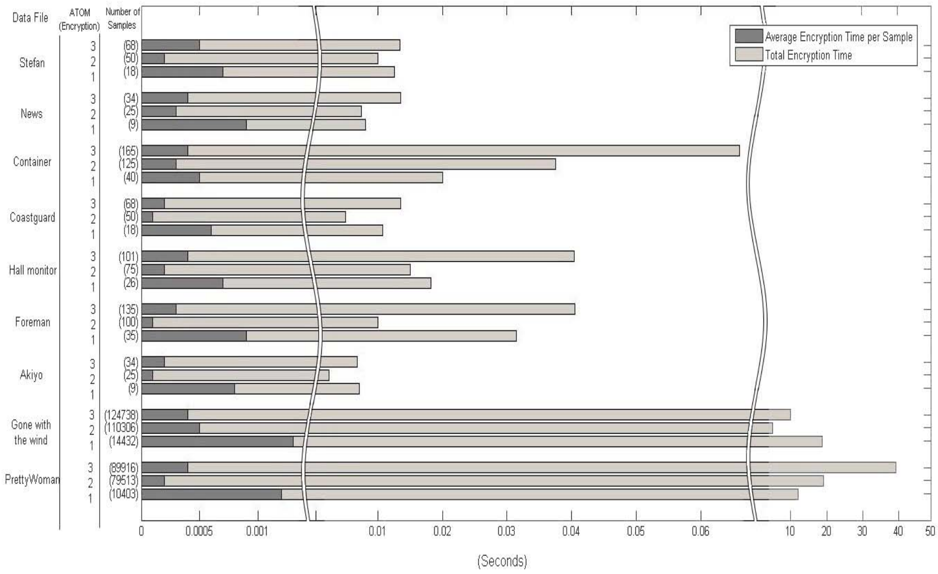

Fig. 10. Encryption time from three encryption methods.

6. Conclusions

In this paper, we proposed two different dropping methods. *Dropping-1* drops the B-frames in an MPEG-4 media file and then the I- and P-frames are moved to the positions of the dropped B-frames. After moving the frames, the Mdat, Stco, and Stts are modified and the media files are sent. *Dropping-2* drops B-frames in real time when transmitting. Either method can lessen the server overload and play high quality media. The *dropping-1* method lessens the server overload by saving a small media data file in the server, but this method needs header modification. *Dropping-2* can be applied without changing any of the existing media header structures or original data. There is little difference between the two methods except for the changes in the media header structures.

We used three different encryption methods to encrypt the macroblocks or motion vectors in VOPs, which extracted video data from MPEG-4 file formats and encrypt VOPs using the DES algorithm. *Encryption-3* offered the strongest encryption results among the methods but required a large amount of data for encryption. The overhead time for the DES encryption algorithm was very slight. The encryption algorithm proposed in this paper has the advantage of not needing any more overhead than a traditional algorithm. This advantage is a strong point for a multimedia streaming service.

In the future, we will need to study dropping methods to reduce server overload for an effective streaming service. Also, we should study DRM further for encryption of transmitted media data.

References

[1] C. C. Bisdikian and B. V. Patel, "Issues on Movie Allocation in Distributed Video-on-Demand Systems," in *Proc. of IEEE International Conf. on Communications*, vol. 1, pp. 250-255, Jun. 1995.

[2] L. Xu and J. Helzer, “Media Streaming via TFRC: An Analytical Study of the Impact of TFRC on User-Perceived Media Quality,” in *Proc. of IEEE INFOCOM*, pp. 1-12, Apr. 2006.

[3] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-Based Congestion Control for Unicast Applications,” *ACM SIGCOMM Computer Communication Reviews*, vol. 30, no. 4, pp. 43-56, Oct. 2000.

[4] “Proposed new text of IPMP FAQ,” *ISO/IEC JTC/SC29/WG11 M8141*, 2002.

[5] J Lacy, N Rump, P Kudumakis, “MPEG-4 Intellectual Property Management & Protection (IPMP) Overview & Application Document,” *ISO/IEC/SC29/WG11/N2614*, 1998.

[6] Ming-Ting Sun, Amy R. Reibman, “Compressed Video over Networks,” *CRC Press*, 2000.

[7] King N. Ngan, Chi W. Yap, Keng T. Tan, “Video Coding for Wireless Communication Systems,” *CRC Press*, 2001.

[8] L. Delgrossi, C. Halstrick, D. hehmann, R. G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, “Media Scaling for Audiovisual Communication with the Heidelberg Transport System,” in *Proc. of 1st ACM international conf. on Multimedia*, pp. 99-104, 1993.

[9] W. Zeng, B. Liu, “Rate Shaping by Block Dropping for Transmission of MPEG Precoded Video over Channels of Dynamic Bandwidth, in *Proc. of 4th ACM international conf. on Multimedia*, pp. 385-393, 1997.

[10] J. Sandvoss, J. Winkler, and H. Witting, “Network Layer Scaling: Congestion Control in Multimedia Communication with Heterogeneous Networks and Receivers,” *Lecture Notes in Computer Science*, vol. 882, pp. 274-293, 1994.

[11] Jae-Gon Kim, Yong Wang, Shih-Fu Chang, Kyeongok Kang, and Jinwoong Kim “Description of utility function based optimum transcoding,” *ETRI Journal*, vol. 27, no. 4, pp. 341-354, 2002.

[12] H. Radha, Y. Chen, K. Parthasarathy, and R. Cohen, “Scalable internet video using MPEG-4,” *Signal Processing: Image Communication*, vol. 15, pp. 95-126, Sep. 1999.

[13] W. Li, J. Ohm, M. V. Schaar, H. Jiang, and S. Li, MPEG-4 “Video verification model ver. 18.0,” *ISO/ IEC/ JTC1/ SC29/ WG11/ N3908*, 2001.

[14] J. R. Ohm, “Advances in Scalable Video Coding,” in *Proc. of the IEEE*, vol. 93, no. 1, pp. 42-56, Jan. 2005.

[15] I. Agi and L. Gong, “An Empirical Study of Secure Mpeg Video Transmissions,” in *Proc. of the Symposium on Network and Distributed System Security*, pp. 137-144, Feb. 1996.

[16] T. B. Maples and G. A. Spanos, “Performance Study of a Selective Encryption Scheme for the Security of Networked, Real-time Video,” in *Proc. of 4th International Conf. on Computer Communications and Networks*, pp. 2-10, 1995.

[17] L. Tang, "Methods for Encrypting and Decrypting MPEG Video Data Efficiently,” in *Proc. of 4th ACM International Multimedia Conf.*, pp. 219-230, 1996.

[18] L. Qiao and K. Nahrstedt, “A New Algorithm for MPEG Video Encryption,” in *Proc. of 1st International Conference on Imaging Science, Systems, and Technology (CISST’97)*, pp. 21-29, 1997.

[19] C. Yuan, B. B. Zhu, Y. Wang, S. Li, Y. Zhong, “Efficient and Fully Scalable Encryption for MPEG-4 FGS,” in *Proc. of IEEE Int. Symp. Circuits and Systems*, vol. 2, pp. 620-623, 2003.

[20] Y. Mao and M. Wu, “A Joint Signal Processing and Cryptographic Approach to Multimedia Encryption,” *IEEE Trans. on Image Processing*, vol. 15, no. 7, pp. 2061-2075, Jul. 2006.

[21] S. Lian, Z. Liu, Z. Ren and Z. Wang, “Selective Video Encryption Based on Advanced Video Coding,” *Lecture Notes in Computer Science*, vol. 3768, pp. 281-290, 2005.

[22] A. Said, “Measuring the strength of partial encryption schemes,” in *Proc. of 2005 IEEE International Conf. on Image Processing (ICIP 2005)*, vol. 2, pp. 1126-1129, 2005.

[23] “QuickTime File Format," *Apple Computer*, 2000.

[24] http://en.wikipedia.org/wiki/MPEG-4

[25] “Information technology-Coding of audio-visual objects-part 1: System ISO/IEC14496-1:2001," *ISO /IEC/SC29/WG11*, 2001.

[26] Data Encryption Standard (DES), *Federal Information Processing Standards Publications (FIPS PUB) 46-3*, 1999.

[27] MPEG4IP, http://mpeg4ip.sourceforge.net/index.php

Dongkyoo Shin received a B.S. in Computer Science from Seoul National University, Korea, in 1986, an M.S. in Computer Science from Illinois Institute of Technology, Chicago, Illinois, in 1992, and a Ph.D. in Computer Science from Texas A&M University, College Station, Texas, in 1997. He is currently a Professor in the Department of Computer Engineering at Sejong University in Korea. From 1986 to 1991, he worked in Korea Institute of Defense Analyses, where he developed database application software. From 1997 to 1998, he worked in the Multimedia Research Institute of Hyundai Electronics Co., Korea as a Principal Researcher. His research interests include XML based middleware, digital right management for multimedia, mobile Internet and ubiquitous computing.

Dongil Shin received a B.S. in Computer Science from Yonsei University, Seoul, Korea, in 1988. He received an M.S. in Computer Science from Washington State University, Pullman, Washington, U.S. A., in 1993, and a Ph.D. from University of North Texas, Denton Texas, U.S.A., in 1997. He was a senior researcher at System Engineering Research Institute, Daejun, Korea, in 1997. Since 1998, he has been with the Department of Computer Engineering at Sejong University in Korea where he is currently an Associate Professor. His research interests include Mobile Internet, Computer Supported Cooperative Work, Distributed Database, Data Mining and Machine Learning.

Jaewan Shin received a B.E. in Computer Engineering from Sejong University, Seoul, Korea, in 2009. He is currently a graduate student in the Department of Computer Engineering, Sejong University, and works as a researcher in the Multimedia & Internet Laboratory at Sejong University. His research interests include Mobile Internet & Computing, U-health Application and Ubiquitous Computing.

Soohan Kim Senior Manager. currently serves as Team Leader of Internet Infra R&D Planning Team of Samsung Electronics Co. HQ. in Rep of Korea. Mr. Kim joined Samsung Electronics Corp. in 1993 as a Junior Engineer in Telecommunication R&D Group. In 1999, he became manager of the R&D part of the Network division, involved in development of the Flight Information System in Inchon Intl' Airport in Korea. In recent years, he was responsible for developing multimedia solution such as DLNA, UX In 2007, he assumed a responsibility of R&D manager of product development in Internet Infra team where his involvement varied in home and office products including IP PBX, Key Phones, Digital Settop Boxes and other products. In 2007. He received his bachelor's degree in material engineering from Chonnam National University(CNU), Gwangju, Korea.

Seungdong Kim received a Ph. D. in System Management Engineering from Sungkyunkwan University, Suwon, Korea, in 2003. He was a senior engineer at Samsung Electronics, Korea, in 2003. Since 2007 ,he has been with Coporate Reseach Center at SK telecom in Korea. His research interests include Intelligent Information Analytics and Contextaware computing, Smart Workplace.