Design of SVC-based Multicasting System Preserving Scalable Security

Kwang-deok Seo, *Member, KIMICS*

***Abstract*— Scalable video coding (SVC) has been standardized as an extension of the H.264/AVC standard. SVC allows straightforward adaptation of video streams by providing layered bit streams. In this paper, we propose a SVC video-based multicasting system preserving scalable security which is able to provide a SVC video service while maintaining information security. In order to maintain information security between a server and a client during all transmission time, the proposed system immediately performs a packet filtering process without decoding with respect to encrypted data received in a routing device, thereby reducing an amount of calculations and latency.**

***Index Terms*— Scalable video coding (SVC), Video Multicasting, Video security, Scalable video adaptation.**

I. INTRODUCTION

The Joint Video Team of the ITU-T VCEG and the ISO/IEC MPEG has recently standardized a Scalable Video Coding (SVC) as a scalable extension of H.264/AVC with the aim at enabling the creation of a compressed bit stream that can be rapidly and easily adapted to fit with the bit-rate of various transmission channels and with the display capabilities and computational resource constraints of various receivers [1]. The scalability in SVC is achieved by taking advantage of the layered approach already known from previous video coding and the scalability is represented in three fundamental types of scalability including spatial, temporal, and quality (SNR) scalabilities. All of these scalability types can be combined to achieve three-dimensional scalability in terms of spatial, temporal, and quality scalabilities. The layers of SVC include one base layer and one or more scalable enhancement layers that can be stacked on top of each other. Each scalable enhancement layer, together with all its dependent lower layers, forms one representation of the video signal at a certain spatio-temporal resolution and quality level.

SVC inherits the structure of H.264/AVC that is divided into two parts, the so-called Video Coding Layer (VCL) and the Network Abstraction Layer (NAL) [1]. In VCL, the scalable video coder creates a coded representation of the source content by employing different techniques to enable spatial, temporal, and quality scalabilities. The NAL encapsulates the data generated by the VCL in NAL units. The NAL unit header carries information about the layer a NAL unit belongs to and its dependencies on other layers. Thus, a substream can easily be extracted at any supported operation point from the complete SVC bit stream.

In a conventional video transmission system utilizing a transcoder which considers various network environments and types of terminals, a new bitstream that satisfies a frame-rate, bit-rate, and video resolution corresponding to conditions of transmission channel and client terminal is generated by the transcoder embedded in a routing or gateway device [2]. Fig. 1 shows a block diagram illustrating a conventional video transmission system employing transcoder for adaptation at the router or gateway. In Fig. 1, a compressed video is encrypted and the encrypted video is loaded on the payload of a RTP (Real-time Transport Protocol) packet to be transmitted to a router. At the router, the arrived encrypted RTP packet is decrypted to recover the original RTP packet. After RTP depacketization, the original compressed video loaded on the payload of RTP packet is transcoded to create a new bitstream that satisfies the required frame-rate, bit-rate, and resolution to be transmitted to a client [3]. After the transcoding, a new RTP packet is generated and it is encrypted again to maintain security during transmission to the client terminal. Since transcoding is possible only in case of approaching to contents of a payload of a received RTP packet, sequential operations of decrypting and decoding in a router are necessary resulting in an increase in an amount of calculations and latency.

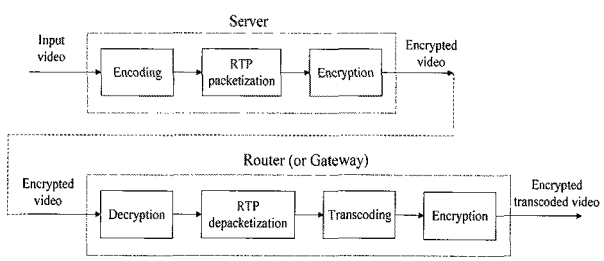


Fig. 1. Block diagram of conventional video transmission system employing transcoder for video adaptation.

Furthermore, it is impossible to securely maintain video information since an original bitstream should be restored

Manuscript received December 11, 2009; revised January 10, 2010; accepted January 28, 2010.

Kwang-deok Seo is with the Computer and Telecommunications Engineering Division, Yonsei University, Gangwon, 220-710, Korea (Email: kdseo@yonsei.ac.kr)

in a router for transcoding. It became clear that to achieve secure video adaptation at an intermediate node within a multimedia delivery network without introducing excessive delay, overhead bandwidth or the need for expensive high speed processors, scalable video is a more promising technique compared to traditional transcoding [4]. Therefore, to achieve consistent information security during all transmission time, we propose an efficient SVC video-based multicasting system which is able to provide a SVC video service while maintaining steady information security.

II. STRUCTURE OF SVC NAL UNIT

An SVC bit stream consists of one or more NAL units. Each NAL unit consists of a header of four octets and the payload byte string as shown in Fig. 2 [5], [6]. In H.264/AVC standard, NAL unit types 14, 15, and 20 have been reserved for future extensions. SVC is using these three NAL unit types. NAL unit type 14 is used for the prefix NAL unit, NAL unit type 15 is used for SVC sequence parameter sets, and NAL unit type 20 is used for coded slice in scalable extension. The scalable NAL unit of type 20 consists of a header of four octets and the payload byte string to encapsulate VCL data. Therefore, NAL unit types 14 and 20 indicate the presence of three additional octets in the NAL unit header extension as shown in Fig. 2. NAL unit header extension part extends the NAL unit header conforming to H.264/AVC by three additional octets and mainly provides the layer decoding dependency information. In the NAL unit header extension part, *TID* (*temporal_id*) indicates the hierarchy between temporal layers for temporal scalability, *DID* (*dependency_id*) denotes the inter-layer coding dependency hierarchy between higher/lower scalable enhancement layers for spatial scalability, and *QID* (*quality_id*) designates the quality level hierarchy of MGS (medium grain scalability) or FGS (fine grain scalability) layers for quality scalability. Therefore, it is possible to distinguish relations between each of the SCV NAL units by exploiting (*DID, TID, QID*) existing in a header of each SVC NAL unit without decoding a bitstream. The NAL unit header is designed to co-serve as the payload header of an RTP payload format. For more details on the syntax and semantics of the SVC NAL unit header, please refer to [5] and [6].

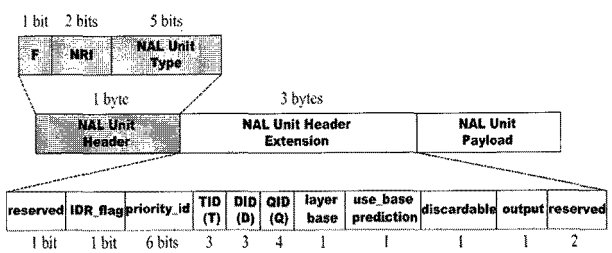


Fig. 2. SVC NAL unit structure.

III. PROPOSED SVC-BASED MULTICASTING SYSTEM

Fig. 3 shows a general service architecture in the case of providing a multicasting service using an SVC video over IP network. Here, it is assumed a multicasting service using an SVC bitstream consists of a single base layer (denoted as B) which is compatible with H.264 and three enhancement layers (denoted as E1, E2, and E3).

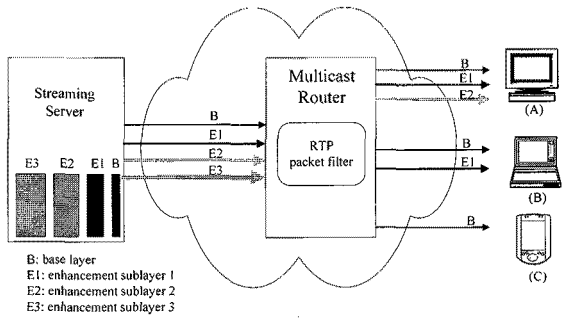


Fig. 3. General service architecture of a SVC-based multicasting service.

When all of NAL units, which configure an SVC bitstream with a total four layers including a base layer, are transmitted to a multicast routing device (or a gateway) via RTP sessions, the routing device extracts only the SVC NAL units corresponding to a suitable resolution, quality, and frame-rate information for each receiving terminal and a channel bandwidth from the all received SVC NAL units, and transmits the extracted SVC NAL units to each client terminal, thereby providing the multicast service.

As an example, in the case where terminal (A) in Fig. 3 is provided with base layer B and upper two enhancement layers of E1+E2, the routing device performs filtering with respect to NAL units corresponding to base layer B and upper two enhancement layers of E1+E2 via an RTP packet filter to transmit the extracted NAL units to terminal (A). As another example, in the case of terminal (B), since capability of a terminal and a channel bandwidth is capable of receiving up to base layer B and enhancement layer E1, the routing device extracts NAL units corresponding to base layer B and enhancement layer E1 to deliver the extracted NAL units to terminal (B). As still another example, in the case of terminal (C), terminal (C) is capable of receiving only the base layer B due to a fact that the terminal (C) lacks in channel bandwidth. Thus the routing device extracts NAL units corresponding to base layer B to transmit the extracted NAL units to terminal (C). Our goal is to achieve the service architecture shown in Fig. 3 while preserving information security during all transmission time. For this purpose, we propose a SVC-based multicasting system as shown in Fig. 4. The principle of the proposed block diagram is based on a method which effectively utilizes a

NAL unit type illustrated in Fig. 2, and (*DID*, *TID*, *QID*) field information.

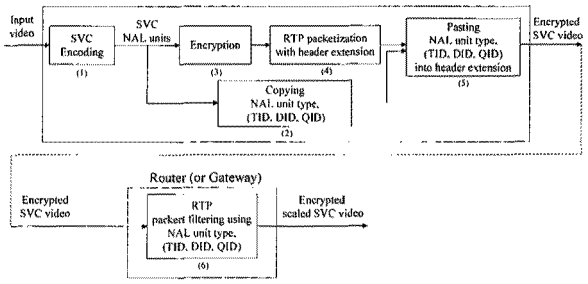


Fig. 4. Block diagram of the proposed SVC-based multicasting system preserving information security.

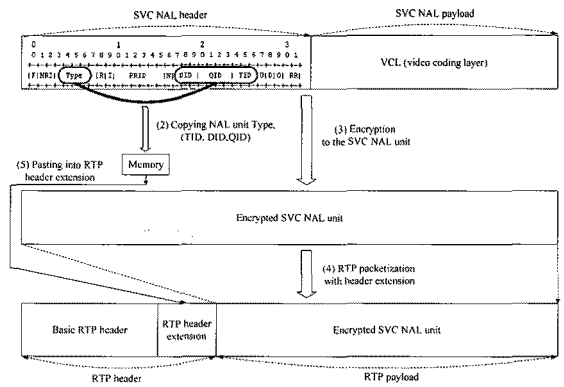


Fig. 5. Detailed procedure to preserve consistent information security described in Fig. 4.

In Fig. 4, the proposed SVC-based multicasting system applies encryption to a SVC NAL unit prior to applying RTP packetization, and thus encryption is not applied to a header portion of an RTP packet. So a routing device may promptly access to information of the header portion in the received RTP packet without decryption process, and it is possible to perform RTP packet filtering at a high speed. Based on this overall procedure, detailed processes for the six steps marked in the block diagram as shown in Fig. 4 are as follows. In step (1), NAL units are generated by SVC encoding. In this instance, NAL unit type and (*DID*, *TID*, *QID*) field information of the NAL unit are copied to a memory in step (2). In step (3), an information encryption module encrypts the generated NAL unit. In step (4), RTP packetization is applied by loading the encrypted video information on a payload of a RTP packet and the RTP header is extended to be able to use the header extension field of the RTP packet. The extracted information such as NAL unit type and (*DID*, *TID*, *QID*) is pasted into the RTP header extension field in step (5). Thereafter, RTP packets including encrypted SVC video are transmitted to the router. In this instance, the router, having received the RTP packet including the encrypted video information, extracts NAL unit type and (*DID*, *TID*, *QID*) from the header extension field of an RTP header. In step (6), packet filtering is performed using the directly available NAL unit type and (*DID*, *TID*, *QID*) to forward only appropriate NAL units to the terminal by satisfying the current channel condition and capabilities of the client terminal. In this process, the router may increase a processing speed since it does not include any processes of decryption, RTP depacketization, video decoding/ encoding, and encryption. Detailed procedure covering the above steps from (2) and (5) which are related to preserve consistent information security is illustrated in Fig. 5. Note that each step from (2) and (5) illustrated in Fig. 5 exactly corresponds to the step marked with the same number in Fig. 4.

IV. ENCRYPTION SCHEMES

Since different multimedia applications require different levels of encryption strength, it is desirable to study two different options for encryption; one that is an accepted, standard full encryption (DES) [7] and one that is less secure (format-compliant shuffling), but offers swifter processing and ease of integration with existing media devices and networks [8]. The full encryption DES provides proof-of–concept for secure video adaptation and also provides a test-bed for exploring real time issues such as delay and errors in streaming and adapting under strict security requirements. The second, format compliant security option was developed as more light-weight encryption which combines the SVC codec along with shuffling at the macroblock level to provide insight into the effectiveness of such encryption.

A. Full Encryption

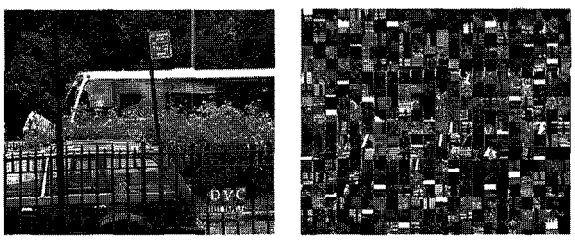
Full encryption partitions each RTP packet into a header field and a data field. It then encrypts the data-carrying field and leaves the header portion untouched. This encryption method requires a detailed study of the underlying video coding method and transportation-layer mechanisms. SVC coding structure includes a video coding layer (VCL), which is designed to efficiently represent the video content, and a network abstraction layer (NAL), which provides "network friendliness" to enable simple and effective customization of the use of the VCL for a broad variety of systems. The first 3 bytes of the NAL unit, which is the NAL unit header, contains three important parameters of the NAL unit: *DID*, *TID*, and *QID*. Whenever intermediate nodes need to perform rate shaping, they extract these three parameters from the RTP header extension of each RTP packet, and then use these three parameters to select and forward RTP packets that are suitable for the current channel condition. Note that with SVC, intermediate nodes only need these three parameters and NAL unit type, not any other information in the NAL unit, for adaptation. Therefore, we propose a full encryption algorithm for SVC: for each RTP packet, we keep the basic RTP header and the RTP header extension unencrypted, and encrypt the rest bits of the RTP packet, as shown in Fig. 5. Keeping the RTP header

and the RTP header extension unaltered will enable adaptation at intermediate nodes, and encrypting the content-carry field in the NAL unit provides end-to-end protection of multimedia content.

In this research, we use an open source free library implementation of the block-cipher Data Encryption Standard (DES) algorithm to encrypt the NAL unit. Other block cipher encryption algorithms, such as the Triple DES, the Advanced Encryption Standard, and the Blowfish, can also be directly applied with little impact on the program structure.

B. Format-compliant Encryption

Format-compliant encryption ensures that the encrypted bit stream maintains the syntax of the compression standard and, therefore, makes the encryption transparent to the networking nodes and the decoder. Syntax compliance enables the direct application of most content adaptation and multimedia signal processing techniques (for example, rate shaping and RTP packetization) to the encrypted bit streams. In addition, it works with the existing communication and networking protocols and can be easily employed. A simple example is to shuffle the multimedia bit stream after compression, and the basic shuffling unit can be 16x16 macroblock, 8x8 DCT block, (Run, Length) codewordes, etc. This encryption method is simply a reorganization of codewords within a compressed bitstream with the structure information (header/marker, etc.) intact. Therefore, it incurs no bit overhead and has low computation complexity. The major drawback of shuffling-based encryption is its relative vulnerability to attacks when compared with full encryption. This method is suitable for applications whose purpose of encryption is quality degradation and reduction of the entertainment value, and offers a light-weight solution to reduce the complexity and enable the real-time play of multimedia.



(a) (b)

Fig. 6. Reconstructed images with and without deshuffling key: (a) with the correct deshuffling key, (b) without deshuffling key.

In this research, we select the shuffling-based format-compliant encryption due to its low computation complexity and little overhead introduced to compression efficiency. In addition, we select macroblock as the basic shuffling unit to ensure its compliance to SVC syntax, and we shuffle the macroblocks within one NAL unit so that encryption will not interfere with rate adaptation at the intermediate network routers. If the SVC decoder has the correct decryption key and deshuffles the bit stream correctly, then the decoder can correctly reconstruct the original frame, as shown in Fig. 6(a). Otherwise, without the correct decryption key, the decoder can still correctly decode the compressed bit stream, but the reconstructed frame does not convey physically meaningful information of the video content, as shown in Fig. 6(b).

V. EVALUATION OF THE PROPOSED MULTICASTING SYSTEM

It is apparent that the proposed multicasting system definitely maintains information security while transmitting secured SVC video over IP networks. By adopting the proposed SVC-based multicasting system preserving information security, we can also avoid subsequent decryption and reencryption processes at the multicast router as shown in Fig. 4. However, the conventional multicasting system requires subsequent decryption and reencryption processes for secure video adaptation at the multicasting router. The basic sequential operations of the conventional multicasting router include decryption for RTP packets, RTP packet filtering for adaptation for the target client's requirements by using ADTE (adaptation decision taking engine), and encryption for the filtered RTP packets, as shown in Fig. 7.

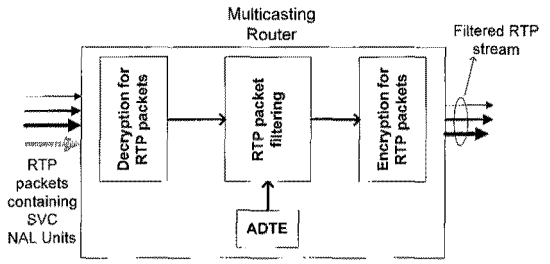


Fig. 7. Basic sequential operations of the conventional multicasting router.

For the case of SVC-based multicast, a number of clients could be simultaneously connected to the multicasting router to receive proper video service. Therefore, the delay time and CPU usage required for processing incoming encrypted RTP packets at the router are important aspects to evaluate the performance of the proposed multicasting system. In addition to the SVC-based multicast employing the proposed method, the basic processing of the conventional multicasting router shown in Fig. 7 was implemented to serve as a reference. This processing of the conventional multicasting router enables us to measure how much basic processing effort the subsequent decryption and encryption steps will induce, which have to be performed by any basic multicasting router as shown in

Fig. 7. As an encryption tool, DES is employed for encrypting all the generated SVC NAL units at the server.

The following quantitative evaluation shows the impact of in-network adaptation and decryption/ encryption on the transmission delay. The video sequences used for the evaluations and their layer configurations are described in Table I. For the encoding of the quality refinement layers, MGS was used and the GoP size of 32 was applied for all the video sequences.

TABLE I
VIDEO SEQUENCES USED FOR EVALUATION

Video Sequence	Base layer	First spatial enh. layer (quality enh.)	Second spatial enh. layer (quality enh.)	Bit-rate in Kbps
City1	QCIF@15Hz (3 MGS)	CIF@30Hz (3 MGS)	None	652
Mobile1	QCIF@15Hz (3 MGS)	CIF@30Hz (3 MGS)	None	735
Foreman1	QCIF@15Hz (3 MGS)	CIF@30Hz (3 MGS)	None	843
City2	CIF@30Hz (3 MGS)	4CIF@30Hz (3 MGS)	None	1287
Mobile2	CIF@30Hz (3 MGS)	4CIF@30Hz (3 MGS)	None	1452
Foreman2	CIF@30Hz (3 MGS)	4CIF@30Hz (3 MGS)	None	1609
City3	QCIF@15Hz (1 MGS)	CIF@30Hz (2 MGS)	4CIF@30Hz (3 MGS)	2081
Mobile3	QCIF@15Hz (1 MGS)	CIF@30Hz (2 MGS)	4CIF@30Hz (3 MGS)	2242
Foreman3	QCIF@15Hz (1 MGS)	CIF@30Hz (2 MGS)	4CIF@30Hz (3 MGS)	2575

We compare the conventional multicasting router (to be denoted as the "conventional router" hereafter) and the router used in the proposed multicasting system (to be denoted as the "proposed router" hereafter). The major role of the routers is to perform packet filtering for video adaptation. We evaluated delay time by implementing the two routers on the 2.6 GHz Quad-core Pentium PC with 1 GB memory running Windows XP. For all measurements, 20 clients were receiving the same content from the streaming server via the two routers. The first 100 seconds were removed from the results data sets, because in the startup phase the number of clients is not constant (i.e., clients are being started one after the other in this time frame). The delay between the streaming server and the client is measured on a picture-by-picture basis. So after fully sending and receiving the last NAL unit of a picture, the timestamp for the picture is recorded. This is done for each stream served by the streaming server and on each client. The adaptation process was enabled for all routers, but the adaptation decision was selected in such a way that all packets are passed through. This would be the worst case scenario, because all of the data has to be handled by the router and transmitted to the client. This "pass-through" operation for all incoming packets is required in order to achieve correct and fair processing time-delay measurements. To illustrate how the bit-rate of the video is responsible for the processing delay time of the two routers, three representative video sequences have been selected from Table I, which are *City1*, *Foreman1*, and *Mobile2*.

Fig. 8 shows the induced delay for the *City1* sequence with an overall bit-rate of about 13 Mbps for all 20 clients. Fig. 9 depicts the delay for *Foreman1* (around 17 Mbps), and Fig. 10 shows the delay for *Mobile2* (around 29 Mbps). The graphs show that each router processing has a very discriminative form of cumulative distribution function (CDF) for the delay, and also with noticeably different mean values and standard deviations. By comparing conventional router and the proposed router, it becomes evident that the additional processing effort by the conventional router is due to the subsequent decryption and encryption. However, the proposed router does not require any decryption and encryption overheads. The delays incurred at the conventional router are approximately 7 to 8 times larger than those of the proposed router. The results in Fig. 8-10 reveal that the delay distribution is also directly dependent on the bit-rate of the video: higher bit-rates lead to increases of both delay and jitter. Note that jitter increases significantly for higher bit-rates, as indicated by the increase in standard deviation. Table II compares the mean and standard deviation of the measured delay times for the three statistical results shown in Fig. 8-10.

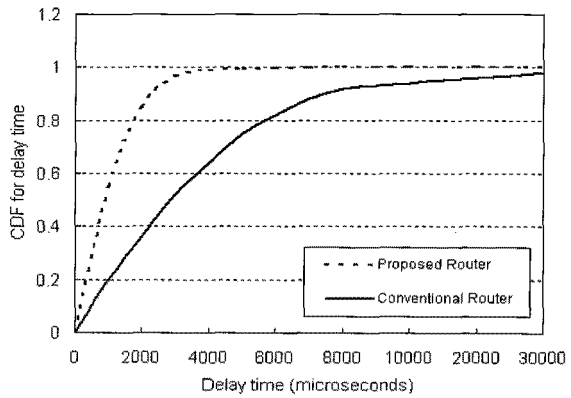


Fig. 8. CDF for delay time of *City1* sequence.

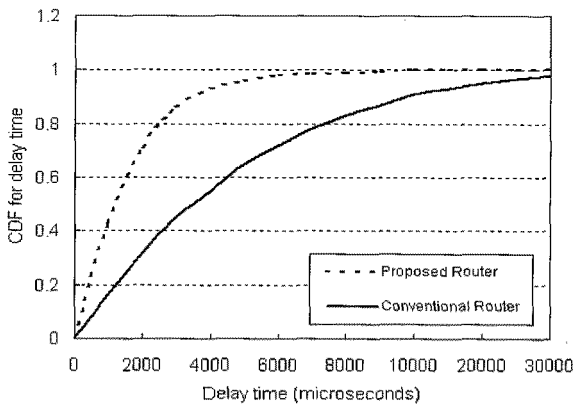


Fig. 9. CDF for delay time of *Foreman1* sequence.

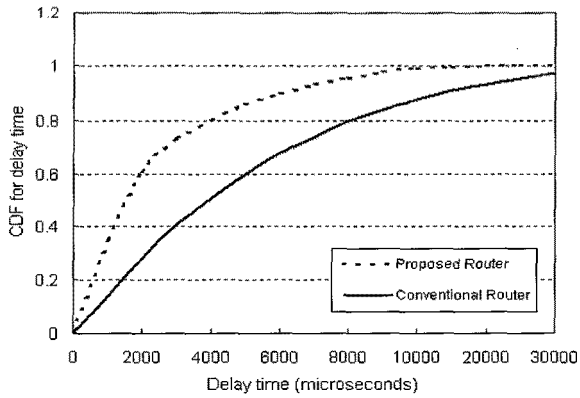


Fig. 10. CDF for delay time of *Mobile2* sequence.

TABLE II
COMPARISON OF MEAN AND STANDARD DEVIATION OF THE MEASURED DELAY TIMES

	City1		*Foreman1*		*Mobile2*	
	Mean (μs)	Std.Dev. (μs)	Mean (μs)	Std.Dev. (μs)	Mean (μs)	Std.Dev. (μs)
Proposed Router	940	553	1243	921	1765	2132
Conventional Router	7131	2812	9245	4213	12152	6375

In addition to the mean and standard deviation, CPU consumption also increases with the bit-rate, but rather modestly. This increase is mainly due to the higher media data throughput. CPU usage results shown in Fig. 11 are consistent with the delay distribution results. While the proposed multicasting system induce a little more computational load at the multicasting server, a significant number of parallel streams (20 simultaneous connections corresponding up to 29 Mbps of throughput) can be handled in real-time on a Pentium PC-based router system. If the number of the simultaneous clients connected to the router increases to more than 20, the effect of the proposed multicasting system in terms of processing delay and CPU usage at the router would become much more significant when compared to the conventional approach.

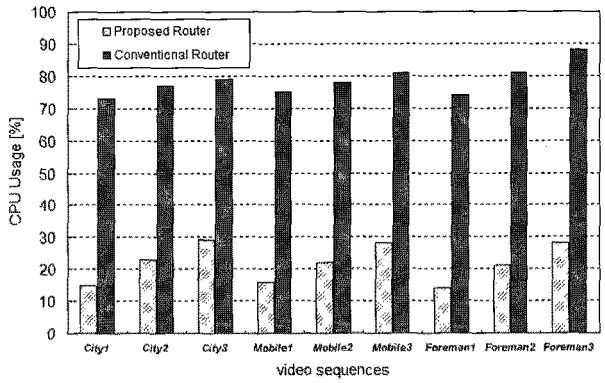


Fig. 11. Comparison of average CPU usage.

VI. CONCLUSIONS

In this paper, we have proposed a SVC video-based multicasting system preserving scalable security which is able to provide a SVC video service while maintaining information security. The proposed system performs a packet filtering process without decoding with respect to encrypted data received in a routing device, thereby reducing an amount of calculations and latency. By employing the proposed system, it is possible to transmit SVC NAL units from a server in a multicasting service to a client with steady information security being maintained and with light computational load.

ACKNOWLEDGMENT

This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (KRF-2008-331-D00378).

REFERENCES

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits and Systems for Video Technol.*, vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[2] A. Vetro, C. Christopoulos, and S. Huifang, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine,* vol. 20, no. 2, pp. 18- 29, Mar. 2003.

[3] C. Gentry, A. Hevia, R. Jain, T. Kawahara, and Z. Ramzan, "End-to-end security in the presence of intelligent data adapting proxies: the case of authenticating transcoded streaming media," *IEEE Journal on Selected Areas in Communications,* vol. 23, no. 2, pp. 464- 473, Feb. 2005.

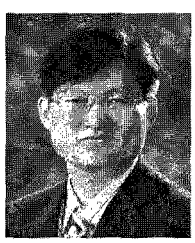
[4] S. Wee, and J. Apostolopoulos, "Secure scalable streaming enabling transcoding without decryption," *IEEE Int. Conf. Image Process.*, 2001.

[5] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M Wien, "Joint draft 11 of SVC amendment," *Joint Video Team*, Doc. *JVT-X201*, Geneva, Switzerland, July 2007.

[6] S. Wenger, Y. Wang, and T. Schierl, "RTP payload format for SVC video," IETF Internet Draft: *draft-ietf-avt-rtp-svc-18.txt,* Mar. 2009.

[7] E. Lin, A. Eskicioglu, and R. Lagendijk, "Advances in Digital Video Content Protection," *Proceedings of the IEEE,* vol. 93, no. 1, pp.171-183, Jan. 2005.

[8] J. Wen, M. Severa, W. Zeng, M. Luttrell, and W. Jin, "A format-compliant configurable encryption frame work for access control of video", *IEEE Trans. Circuits systems for video technol.*, vol. 12, no. 6, pp. 545-557, June 2000.



Kwang-deok Seo received the B.S., M.S. and Ph.D. degrees in Electrical Engineering from KAIST, Daejeon, Korea, in 1996, 1998, and 2002, respectively. From Aug. 2002 to Feb. 2005, he was with LG Electronics. Since March 2005, he has been a Faculty Member in the Computer and Telecommunications Engineering Division, Yonsei Univ., Gangwon, Korea, where he is an associate professor. His current research interests include digital video broadcasting, mobile IPTV, scalable video coding, and protocol design for scalable video transport. He is a member of IEEE, KICS, KIMICS, and IEICE.