

SVC 비디오 전송을 위한 RTP 타임스탬프 자동 생성 방법

종신회원 서 광 덕*, 정회원 정 순 흥***, 김 재 곤**, 유 정 주***

Automatic RTP Time-stamping Method for SVC Video Transmission

Kwang-deok Seo* *Lifelong Member,*

Soon-heung Jung***, Jae-gon Kim**, JeongJu Yoo*** *Regular Members*

요 약

본 논문에서는 인터넷과 같은 IP망에서 SVC 비디오를 RTP 패킷화하여 전송할 때 RTP 패킷의 헤더에 기록될 타임스탬프 정보를 자동으로 생성할 수 있는 새로운 알고리즘을 제안한다. H.263, MPEG-4, H.264 등의 단일 계층 비디오 압축 방식과는 달리 SVC는 하나의 비트스트림으로 시간적, 공간적, SNR 스케일러빌리티를 동시에 제공하기 위해 다중 계층 부호화에 의해 SVC 비트스트림을 생성하게 된다. 특히, 시간적 스케일러빌리티 제공을 위해 계층적 B-픽처 (hierarchical B-picture) 예측 구조를 채택하는 관계로 화면의 부호화 (또는 전송) 순서와 디스플레이 순서가 서로 일치하지 않는다. 따라서, 각 화면에 대한 RTP 패킷 헤더에 기록될 타임스탬프 값의 증가치가 불규칙적으로 나타나게 된다. SVC 비디오 전송 환경에서 RTP 타임스탬프 값을 실시간으로 효과적으로 생성해 줄 수 있는 방법이 현재까지 제안된 적이 없다. 따라서, 본 논문에서는 SVC 비디오 전송 환경에서 NAL unit 헤더에 기록되는 *TID* (Temporal ID) 필드를 이용하여 RTP 타임스탬프 값을 자동적으로 생성하는 방법을 제안한다.

Key Words : SVC Video Transport, RTP Packetization, RTP Timestamp

ABSTRACT

In this paper, we propose a novel algorithm to automatically generate an RTP timestamp value that is required for the RTP packetization in order to transmit SVC video over various IP networks such as Internet. Unlike the conventional single layer coding algorithms such as H.263, MPEG-4 and H.264, SVC generates a multi-layered single bitstream which is composed of a base layer and one or more enhancement layers in order to simultaneously provide temporal, spatial, and SNR scalability. Especially, in order to provide temporal scalability based on hierarchical B-picture prediction structure, the encoding (or transmission) and display order of pictures in SVC coding is completely decoupled. Thus, the timestamp value to be specified at the header of each RTP packet in video transmission does not increase monotonically according to the display time instant of each picture. Until now, no method for automatically generating an RTP timestamp when SVC video is loaded in a RTP packet has been introduced. In this paper, a novel automatic RTP timestamp generation method exploiting the *TID* (temporal ID) field of the SVC NAL unit header is proposed to accommodate the SVC video transmission.

* 본 연구는 정보통신연구진흥원의 IT신성장동력핵심기술개발 사업의 일환으로 수행하였음. [2005-S-103-02, 통방융합 환경에서의 유비쿼터스 콘텐츠 서비스(UCA) 기술]

* 연세대학교 컴퓨터정보통신공학부 (kdseo@yonsei.ac.kr), ** 한국항공대학교 항공전자 및 정보통신공학부 (jgkim@kau.ac.kr)

*** 한국전자통신연구원 방통미디어연구부 (zeroone@etri.re.kr), (jjyoo@etri.re.kr)

논문번호 : KICS2007-12-584, 접수일자 : 2007년 12월 24일, 최종논문접수일자 : 2008년 5월 26일

I. 서 론

H.264/AVC의 확장형 부호화 기술인 SVC는 기존의 MPEG-2, MPEG-4등에서 시도한 계층 부호화 기반의 스케일러빌리티가 갖는 문제점인 낮은 압축 효율, 복합 스케일러빌리티 지원불가, 높은 구현 복잡도의 문제를 한꺼번에 해결하기 위하여 개발된 새로운 확장형 부호화 기법이다⁶⁾. 현재 JVT (Joint Video Team)에서 국제표준화 작업이 마무리 단계이며 ISO/MPEG에서 최종적으로 AVC Amendment 형태로 승인 될 예정이다.

SVC는 여러 개의 비디오 계층 (layer)을 하나의 비트열로 부호화한다. SVC의 계층은 하나의 기본계층 (base layer) 과 기본계층 위에 연속적으로 쌓을 수 있는 확장계층 (scalable enhancement layer)으로 구성된다. 각 확장계층은 하위 계층 정보를 기반으로 각각에게 주어진 최대의 비트율 (bit-rate), 화면율(frame rate), 해상도 (resolution)까지 표현할 수 있다. SVC에서는 확장계층을 연속적으로 많이 쌓을 수록 다양한 비트율, 화면율, 해상도의 지원이 가능하므로, 이종의 (heterogeneous) 망 환경에서 발생하는 대역폭의 다양성 문제, 수신 단말기 성능과 해상도의 다양성 문제, 콘텐츠 소비자의 다양한 선호도 문제 등을 복합적으로 해결할 수 있는 UMA (Universal Multimedia Access) 환경의 멀티미디어 콘텐츠 서비스에 적합한 부호화 기술이다⁷⁾.

SVC 부호기의 VCL (Video Coding Layer) 계층에서는 기본계층 부호화 정보와 확장계층의 스케일러빌리티 부호화 정보를 슬라이스 단위로 생성하게 된다. 생성된 슬라이스는 네트워크를 통해 전송되기 위하여 NAL (Network Abstraction Layer) 계층에서 NAL unit 으로 만들어지게 된다. SVC NAL unit 들로 구성된 SVC 비트스트림은 IP (Internet Protocol) 망을 통하여 전달이 되기 위하여 RTP 패킷화 과정을 거치게 되는데, RTP 패킷화 과정에서는 다른 종류의 미디어 정보와의 동기화 (synchronization)를 지원하기 위해서 RTP 헤더의 RTP 타임스탬프 (timestamp) 정보를 수신 측에 반드시 전송하여야 한다³⁾. 예를 들면, SVC 비디오를 AAC 등의 오디오와 함께 서비스할 경우 수신측에서의 비디오와 오디오 간의 입술동기화 (lip synchronization)를 지원하기 위해서 RTP 타임스탬프 정보의 전송은 필수적이다^{1),2)}. 만약 SVC file format 기반의 SVC 콘텐츠를 활용하여 스트리밍과 같은 비디오 서비스를 실시할 경우에는 파일 포맷

헤더에 기록되어 있는 CTS (composition time stamp) 정보를 기반으로 RTP 타임스탬프 값을 생성하여 RTP 패킷을 만들 수 있다⁹⁾. 그러나, 파일 포맷 규격을 따르는 콘텐츠 파일에 저장되지 않은 SVC 비트스트림을 이용하여 비디오 전송을 할 경우에는 RTP 타임스탬프 값을 별도로 제공하는 방법이 필요하다. 또한 라이브 비디오 전송 환경에서는 카메라를 통해 입력된 영상 신호를 SVC 부호기로 압축 한 후에 생성된 NAL unit을 RTP 패킷에 실시간으로 실어서 전송해야 하기 때문에 RTP 패킷 헤더에 기록되는 타임스탬프 정보를 자동으로 생성하는 별도의 알고리즘이 필수적이다. 그동안 JVT에서 SVC 비디오 압축 표준화 작업은 활발히 진행되어 마무리 단계이지만 앞으로는 SVC 압축 표준을 기반으로 SVC 비디오 전송에 필요한 제반 규격에 관한 개발이 절실한 상황이다.

그림 1의 블록도는 이러한 SVC 비디오 전송 환경에서 SVC 비디오 비트스트림을 생성하여 실시간으로 RTP 패킷을 만들 때 본 논문에서 제안하는 방법이 적용되는 과정을 보인다. 제안된 방법에 의해 현재 RTP 패킷에 실리게 되는 SVC NAL unit 에 대한 타임스탬프 정보를 자동으로 생성하게 되고 RTP 헤더에 그 값을 기록할 수 있다.

그림 2는 SVC 비디오가 실리게 되는 RTP 패킷의 구조를 나타낸다³⁾. RTP 패킷은 크게 헤더부분과 페이로드 (payload) 부분으로 나뉘어 지는데, 헤더부분에는 32비트 크기의 타임스탬프 구간이 존

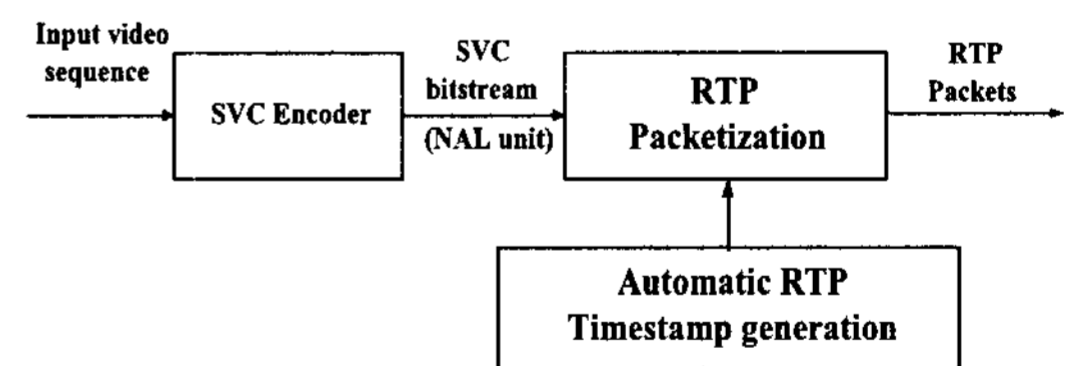


그림 1. SVC의 RTP 패킷화를 위한 타임스탬프 자동 생성

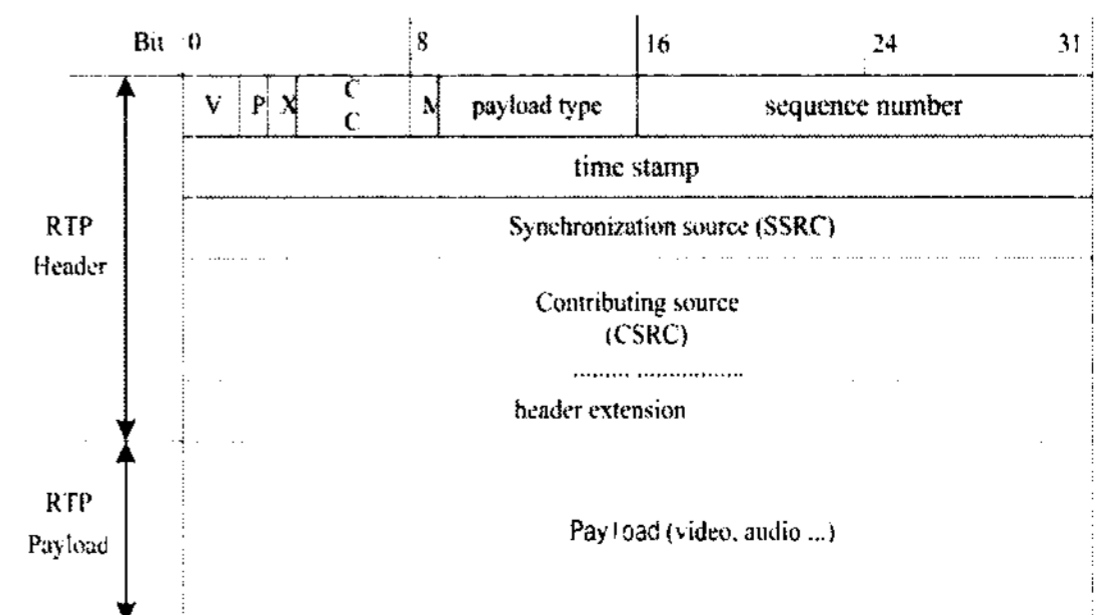


그림 2. RTP 패킷의 구조

재하게 되며 이 구간에 현재 전송하게 되는 SVC NAL unit에 대한 타임스탬프 정보를 기록하게 된다.

본 논문에서는 그림 2의 RTP 패킷 헤더에 기록될 타임스탬프 정보를 자동으로 생성하여 그림 1의 과정을 통해 RTP 패킷을 효과적으로 생성할 수 있는 알고리즘을 제안한다.

II. SVC NAL unit과 복합 스케일러빌리티

2.1 SVC NAL unit 의 헤더 구조

SVC 부호화에 의해 생성된 부호화 정보는 NAL unit으로 비트스트림에 저장되는데, 그림 3에 보이듯이 기본계층에서 생성된 NAL unit과 확장계층에서 생성된 NAL unit의 헤더 구조는 서로 다르다. 기본계층의 NAL unit은 H.264/AVC와 호환이 되어야 하므로 H.264/AVC의 NAL unit 헤더 구조와 동일한 헤더필드 3가지로 구성되며 총 1바이트 크기를 갖는다⁴⁾. 확장 계층에서 생성된 NAL unit 헤더의 경우 기본계층 NAL unit 헤더에 추가적으로 3바이트 크기를 갖는 새로운 헤더 필드가 붙게 된다.

그림 3에서 (*TID*, *DID*, *QID*) 필드는 각각 시간적, 공간적, SNR 스케일러빌리티에서 각 계층간의 시공간적 관련 체계를 나타낸다. 즉, *TID* (*Temporal_ID*)는 시간적 스케일러빌리티를 위한 시간적 계층 (*temporal level*) 간의 체계 (*hierarchy*)를 나타내고, *DID* (*Dependency_ID*)는 공간적 스케일러빌리티의 계층간 예측 (*inter-layer prediction*)에 있어서 상하위 확장계층 간의 종속 체계 (*dependency hierarchy*)를 나타내며, *QID* (*Quality_ID*)는 SNR 스케일러빌리티 지원을 위한 FGS 계층 간의 계층 체계를 나타낸다. *TID*, *DID*, *QID* 값은 모두 0 이상의 정수이며, 이 값들의 조합으로부터 각 NAL unit에 대한 시공간적 계층 관계를 유도해 낼 수 있다. 본 논문의 목적인 타임스탬프를 자동으로 생성하기 위해서 사용되는 정보는 (*TID*, *DID*, *QID*) 필드 중에서 *TID* 정보이다. 본 논문에서는 SVC 압축 후 생성되는 NAL unit의 헤더에 기록되

어 손쉽게 접근할 수 있는 *TID* 값을 이용하여 주어진 NAL unit에 적용될 타임스탬프 값을 자동으로 생성하는 방법을 제안하게 된다.

2.2 SVC의 복합 스케일러빌리티

SVC의 복합 스케일러빌리티에서는 시간적, 공간적 및 SNR 스케일러빌리티를 하나의 SVC 비트스트림으로 통합해서 제공할 수 있다^{6),8)}. 그림 4는 SVC의 복합 스케일러빌리티 부호화를 위한 화면 및 계층 구조의 예를 보인다. 이 그림에서는 SVC 스트림의 시작 부분인 IDR 픽처와 첫번째 GOP에 해당하는 화면만을 보이는데, 하나의 GOP는 16장의 화면으로 구성된다. 이 그림에서 기본계층에서 지원 가능한 화면 해상도는 QCIF이고 공간적 확장계층 (*spatial scalable layer*)에서 지원 가능한 화면 해상도는 CIF이다. 서로 다른 공간적 확장 계층에서 각각 표현이 가능한 화면 해상도를 구분하기 위하여 그림 3의 (*TID*, *DID*, *QID*) 필드 중 *DID* 값을 이용한다. 즉, 그림 4에서 *DID*=0을 갖는 NAL unit은 해상도가 QCIF 인 화면을 의미하며 *DID*=1로 설정된 NAL unit은 해상도가 CIF 인 화면을 나타낸다. 시간적 스케일러빌리티 제공을 위해서 계층적 B-픽처 (*hierarchical B-picture*) 방식이 적용되며 지원 가능한 화면율을 표시하기 위하여 (*TID*, *DID*, *QID*) 필드 중 *TID* 값을 이용한다. 그림 4에서 *TID* 값은 사각형으로 표시된 각 화면의 가운데 부분에 숫자로 표시가 되어 있다. *TID*=0 인 키 픽처 (*key picture*)만 전송할 경우 1.875 fps (*frame per second*) 까지 화면을 지원이 가능하며, *TID*=1인 B-픽처를 포함하여 전송할 경우 3.75 fps 까지 화면을 지원이 가능하다. 추가적으로 *TID*=2인 B-픽처를 전송할 경우 7.5 fps까지 얻을 수 있으며, *TID*=3과 *TID*=4 로 표시된 B-픽처를 추가적으로 전송하게 되면 각각 15 fps 와 30 fps 까지 화면을 지원이 가능하다. 그림 4의 경우 기본계층에서의 최대 *TID* 값은 3이므로 기본계층에서는 QCIF 규격으로 최대 15 fps까지 지원할 수 있고, 공간적 확장계층의 최대 *TID* 값은 4이므로 CIF 규격으로 최대 30 fps까지 지원할 수 있다. 기본계층과 공간적 확장계층에서 동일한 시점 (*time instant*) 에 해당되어서 서로 같은 *TID* 값을 갖게 될 경우 그림에서 점선으로 표시된 화살표 방향으로 계층간 예측부호화 (*inter-layer prediction*)가 적용될 수 있는데, *DID*=0인 QCIF규격의 기본계층 화면의 해상도가 업샘플링 (*up-sampling*) 되어 *DID*=1인 CIF 규격의

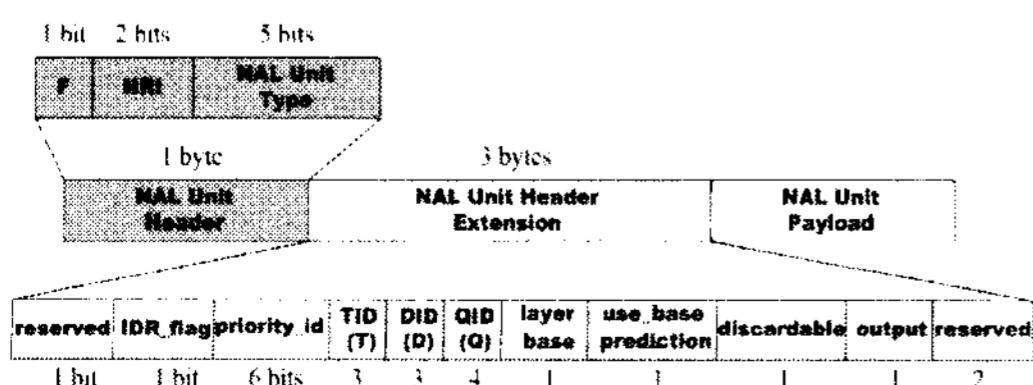


그림 3. SVC의 기본계층과 확장계층에서 사용되는 NAL unit의 헤더 구조⁵⁾

확장계층 화면의 예측부호화에 활용이 된다. 한편, 이 그림에서는 SNR 스케일러빌리티 지원을 위하여 화면마다 FGS 계층을 하나씩 생성하고 있으므로, 각 FGS 계층의 부호화 정보에 해당되는 NAL unit은 모두 QID =1로 설정이 된다.

III. 제안된 RTP 타임스탬프 자동생성 방법

그림 4에 나타난 SVC 복합 스케일러빌리티 부호화를 위한 화면 구조에서 GOP 내부의 1~16번 화면이 모니터에 디스플레이 되는 순서를 고려하여 RTP 패킷 헤더에 기록해야 할 타임스탬프 값은 그림 5와 같다. 그림 4의 화면 및 계층구조에서는 TID의 최대값이 4이므로 최대 30 fps까지 지원이

가능하다. 비디오 화면에 대한 RTP 타임스탬프 값을 생성할 때 사용되는 샘플링 클럭 (sampling clock) 은 일반적으로 90 KHz 클럭을 사용하도록 표준으로 정해져 있다^[3]. 따라서, 30 fps 까지 지원되는 비디오의 경우 화면간 클럭 간격 (inter-frame clock interval) 은 다음 식에 의해 3000 clocks/frame 으로 계산된다.

$$\begin{aligned} \text{Inter-frame_Clock_Interval} &= \frac{90,000 \text{ Hz}}{\text{Max_FR}} \\ &= \frac{90,000 \text{ clocks/s}}{30 \text{ frames/s}} = 3,000 \text{ clocks/frame} \end{aligned} \quad (1)$$

그림 5에서는 화면번호 0인 IDR 픽처가 포함된 경우인데 IDR 픽처에 대한 초기 타임스탬프 값을 0

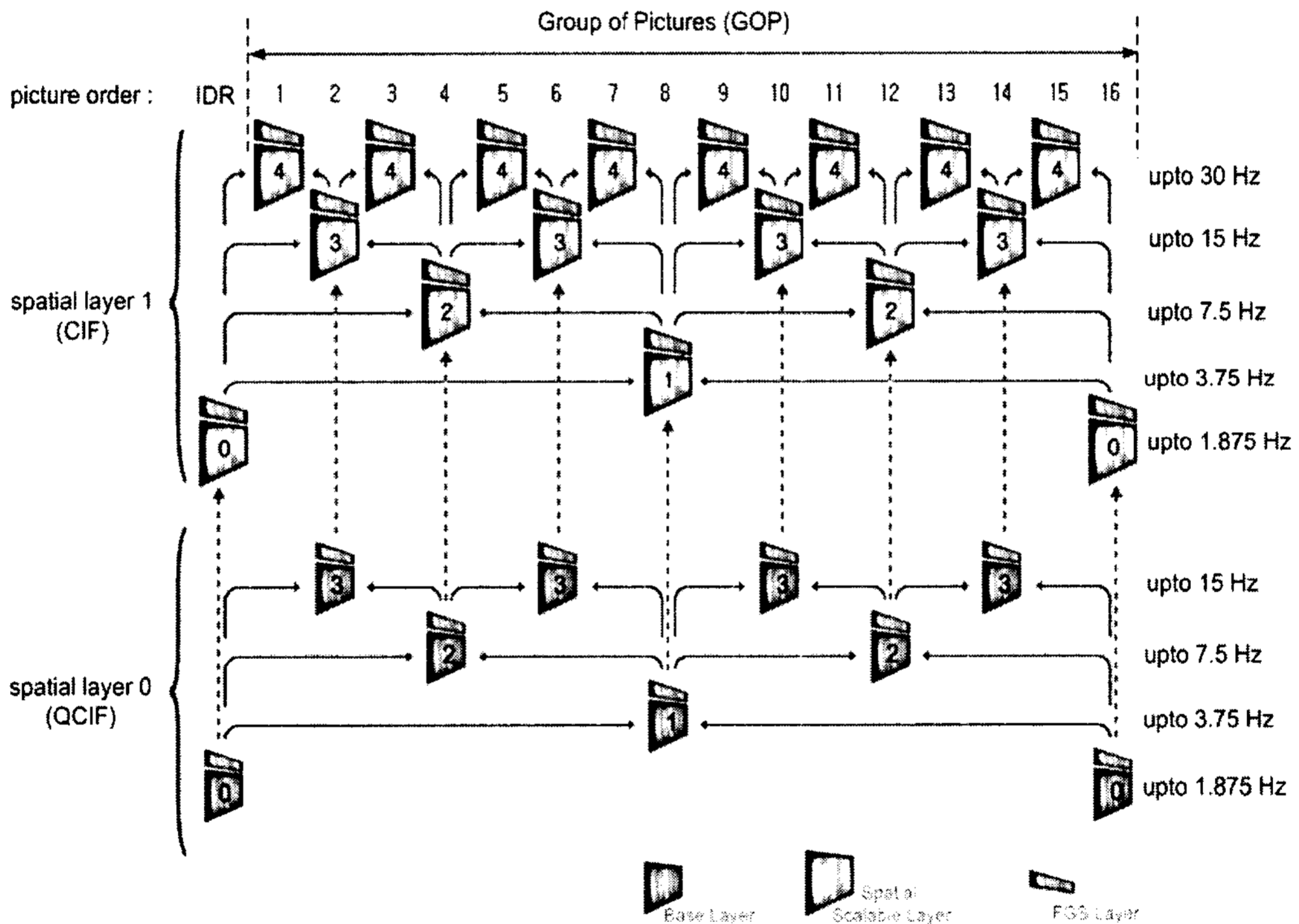


그림 4. SVC의 복합 스케일러빌리티

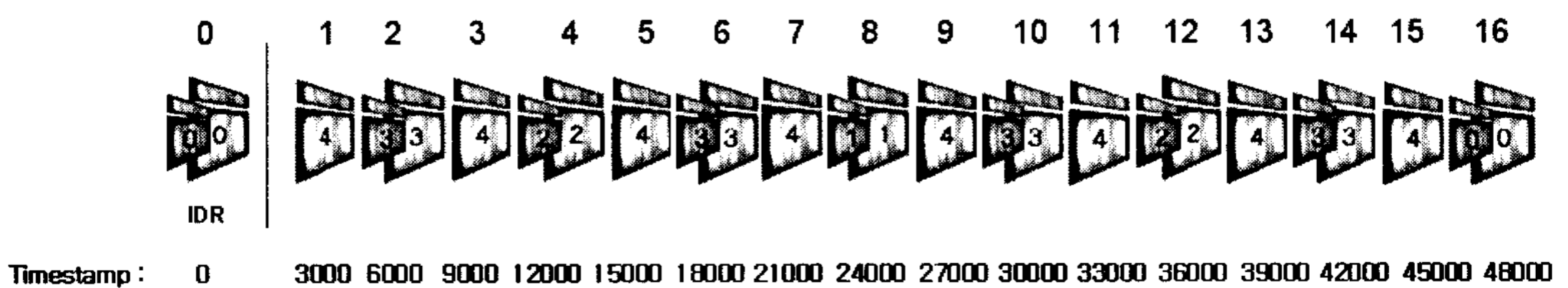


그림 5. 화면의 디스플레이 순서와 각 화면에 대한 타임스탬프 값

으로 설정하였다. 만약 IDR 픽처에 대한 초기 타임스탬프 값이 보안을 목적으로 임의의 값 (random value) 으로 초기화 되면 이 임의의 값에 대해 3000만쯤 클럭 (clock) 수를 단조 증가시킨 값을 각 화면의 타임스탬프 값으로 적용하면 된다.

이처럼 각 화면의 디스플레이 순서 별로 타임스탬프 값을 설정하는 것은 매우 간단하지만, 문제는 화면의 디스플레이 순서와 동일하게 SVC 부호화 및 전송이 수행되지 않는다는 것이다. SVC에서는 시간적 스케일러빌리티를 제공하기 위하여 계층적B-픽처 예측 부호화를 적용하기 때문에 화면의 디스플레이 순서와 부호화 순서 (또는 전송순서)는 서로 다르다. 따라서, 그림 5에 보이는 타임스탬프 값의 규칙적인 변화를 기반으로 타임스탬프를 각 화면 간격마다 3000씩 증가시켜서 적용할 수 없기 때문에, 각 화면별 부호화 순서와 디스플레이 순서가 상이함을 고려하는 새로운 방법의 제안이 필요하다.

그림 6은 그림 4의 SVC 부호화 구조에서 화면들을 부호화 순서 (또는 전송순서) 대로 나열한 후 그림 5에서 설정한 각 화면에 해당하는 타임스탬프 값을 각 화면마다 매핑한 결과이다. 그림 6에 보듯이 각 화면마다 설정된 타임스탬프 값은 화면마다 매우 불규칙적인 변화를 보이고 있다. 따라서 각 화면에 할당될 타임스탬프 값을 미리 계산하여 별도의 파일로 저장해 놓고 각 화면에 해당하는 NAL unit을 RTP 패킷에 실을 때 이미 파일에 저장되어 있는 타임스탬프 값을 호출하여 적용해야 한다. 그러나 이 방법에서는 추가적으로 타임스탬프 값을 저장하는 파일 제작 작업이 요구되고 또한 파일 저장을 위한 별도의 메모리가 필요하게 되어 화면의 개수가 매우 많을 경우 파일 저장을 위한 메모리 요구량이 커지게 되어 매우 비효율적이다. 따라서, SVC 비트스트림 기반의 비디오 전송 환경에서는 즉각적으로 각 NAL unit에 해당되는 타임스탬프 값을 생성하는 새로운 알고리즘을 필요로 한다.

이러한 문제점을 해결하기 위하여 본 논문에서는

SVC 비디오 전송의 시작과 동시에 각 SVC 화면에 대한 타임스탬프 값을 자동적으로 생성할 수 있는 새로운 방법을 고안하고자 한다. 고안된 방법에서는 그림 3에 제시된 NAL unit 헤더에 포함되어 있는 TID 값 만을 이용하여 동작하므로 별도의 추가적인 정보가 필요하지 않아 동작구조가 간단하며 계산 복잡도 또한 낮다.

제안되는 방식을 설명하기 위해서는 *TL_Group* (temporal level group)과 *TL_Group_Size* 라는 새로운 용어를 사용하게 되는데, *TL_Group* 은 하나의 GOP 내에서 동일한 TID 레벨을 갖는 화면들의 묶음을 의미하며, *TL_Group_Size* 는 동일한 *TL_Group* 에 속하는 화면의 개수를 의미한다. 이 용어들을 그림 6에 적용할 경우 그림 7에 제시되어 있는 결과로 *TL_Group*과 *TL_Group_Size* 가 결정된다. 그림 7의 경우 TID 값이 0인 16번 화면이 독립된 *TL_Group* 을 구성하며 *TL_Group_Size* 는 1이 된다. TID 값이 1인 8번 화면도 독립된 *TL_Group*이 되며 *TL_Group_Size* 는 1이 된다. TID 값이 2인 4번과 12번 화면들이 독립된 *TL_Group*을 형성하며 *TL_Group_Size* 는 2가 된다. TID 값이 3인 2, 6, 10, 14번 화면들도 독립된 *TL_Group*이 되며 *TL_Group_Size* 는 4가 된다.

TID 값이 4인 1, 3, 5, 7, 9, 11, 13, 15번 화면들도 독립된 *TL_Group*이 되며 *TL_Group_Size* 는 8이 된다.

지금부터, 이러한 *TL_Group*과 *TL_Group_Size* 를 바탕으로 각 화면마다 할당될 타임스탬프를 자동으로 생성하게 되는 수식을 제안하고자 한다.

먼저, GOP 내에서 첫 번째로 부호화 되는 키 픽처에 해당되는 화면의 타임스탬프 TS_{Key_Pic} 값을 계산하기 위하여 다음의 수식이 제안된다.

$$TS_{Key_Pic}(T_{MAX}) = IDR_TS + Clock_Int \times 2^{T_{MAX}} \times GOP_Num \tag{2}$$

식 (2)에서 T_{MAX} 는 부호화 시 현재 GOP 내 화

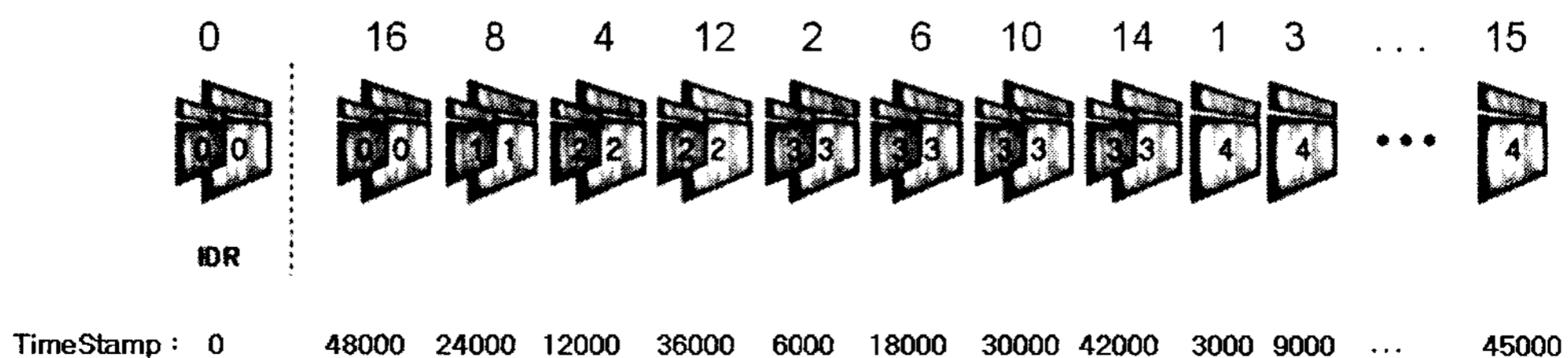


그림 6. 화면의 부호화 또는 전송 순서에 따른 타임스탬프 값

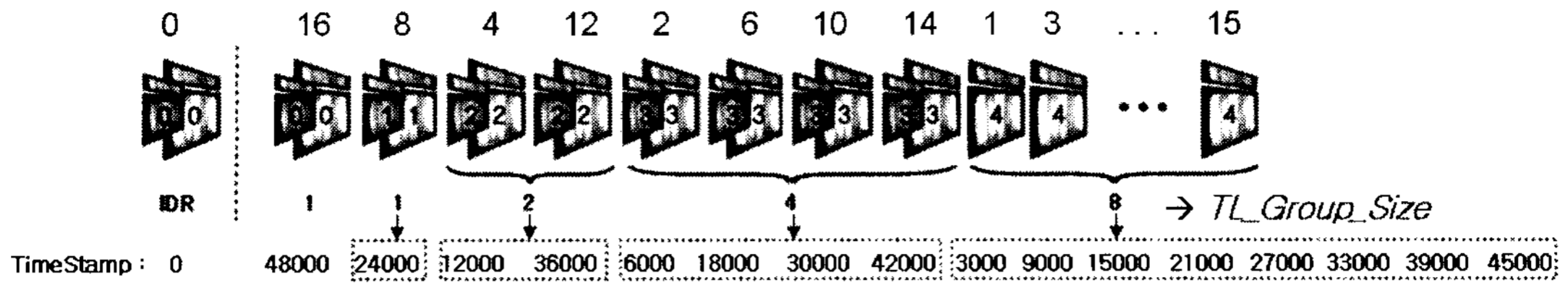


그림 7. *TL_Group* (점선 박스 묶음으로 표시) 과 *TL_Group_Size* 설정

면의NAL unit 들에 할당될 *TID* 값 중 최대의 *TID* 값을 의미하고 *Clock_Int* 는 화면간 타임스탬프 값의 시간 간격을 의미하는데 식 (1)을 통해 계산되어진다. *IDR_TS* 는 SVC 스트림의 첫번째 화면인 IDR 픽처가 갖게 되는 타임스탬프 값을 의미하며 *GOP_Num* (≥ 1) 은 SVC 스트림이 포함하는 많은 GOP 중에서 현재 전송 중인GOP 의 순서번호를 나타낸다.

GOP 내에서 키 픽처를 제외한 나머지 화면들에 대한 타임스탬프 값을 얻어내기 위한 정형화된 수식을 만들기 위해서 그림 7로부터 일정한 규칙 2가지를 발견할 수 있었다. 첫째, 각 *TL_Group* 의 첫번째 화면이 갖는 타임스탬프 값은 항상 이전 *TL_Group* 의 첫번째 화면이 갖는 타임스탬프 값의 절반이 된다. 둘째, 동일한 *TL_Group* 내에서 첫번째 화면을 제외한 나머지 화면들은 순서대로 첫번째 화면이 갖는 타임스탬프 값의 3배, 5배, 7배, 9배, 11배등 홀수배로 증가하는 타임스탬프 값을 가짐을 알 수 있다. 이러한 두가지 규칙을 바탕으로 키 픽처를 제외한 나머지 화면들에 대한 타임스탬프 값 TS_{Pic} 은 다음 수식을 통해 구해낼 수 있다.

$$TS_{Pic}(T, n) = IDR_TS + \{Clock_Int \times 2^{T_{MAX}} \times (GOP_Num - 1)\} + Clock_Int \times 2^{T_{MAX}-T} \times (2 \times n + 1) \quad (3)$$

식 (3)에서 T ($1 \leq T \leq T_{MAX}$)는 현재 화면의 *TID* 값을 의미하며, n 은 동일한 *TL_Group* 내에서 현재 화면의 순서번호를 나타내며 그 값의 범위는 $0 \leq n < TL_Group_Size$ 이다.

이러한 수식을 바탕으로 SVC 비디오 전송에서 NAL unit 단위로 주어진 화면에 대한 타임스탬프 값을 능동적으로 생성하기 위한 순서도는 그림 8과 같다. 현재 RTP 패킷화 되는 NAL unit 이 IDR 픽처에 속하면서 이 화면 내에서 첫번째 NAL unit에 해당된다면 타임스탬프 값으로 *IDR_TS* 를 설정한

다. *IDR_TS*는 0으로 초기화 하여 할당 할 수도 있으나 보안을 위하여 임의의 값으로 설정될 수 있다. 현재 RTP 패킷화 되는 NAL unit 이 GOP 내에서 처음으로 부호화되는 키 픽처에 속하면서 이 키 픽처 내에서 처음으로 생성된 NAL unit이라면 식 (2)를 이용하여 TS_{Key_Pic} 를 계산하게 된다. 만약, 현재 RTP 패킷화 되는 NAL unit이 IDR 픽처 또는 키 픽처가 아닌 일반 픽처에 속하면서 화면 내부에서 처음으로 생성된 NAL unit이라면 식 (3)을 이용하여 TS_{Pic} 을 계산하게 된다. 이러한 모든 조건들이 만족되지 않는 NAL unit 의 경우에는 각 화면 내에서 처음으로 RTP 패킷화 처리된 NAL unit 에 대해 구해진 기존의 타임스탬프 값을 그대로 적용하여 RTP 패킷 헤더에 기록하면 된다. 이와 같이 하는 이유는 동일한 화면 내에 속하는 모든 NAL unit 들은 동일한 화면 표현 시점 (presentation time instant)에 해당하기 때문이다. 즉, 화면 내의 첫 번째 NAL unit에 할당된 타임스탬프와 동일한 타임스탬프 값을 갖기 때문에 별도의 계산을 추가적으로 할 필요가 없기 때문이다.

IV. 실험 결과

그림 4에 나타나 있는 GOP 화면 구조를 갖는 SVC 비트스트림을 RTP 패킷화하여 전송할 때 제안된 RTP 타임스탬프 자동생성 방법을 적용한다. 그림4의 화면 및 계층 구조에서는 하나의 기본계층과 최대 3개의 확장계층이 존재한다.

이 실험에서는 SVC 전송을 위하여 IETF 표준에서 규정하는 세가지 전송 모드인 SNU 모드 (single NAL unit mode), Non-interleaved 모드, Interleaved 모드 중에서 Non-interleaved 모드를 사용하였다. Non-interleaved 모드에서는 SNU 타입, FU-A 타입, STAP-A 타입 등 총 3가지의 RTP 패킷 타입을 지원한다^[5]. SNU 타입은 하나의 RTP 패킷에 하나의 NAL unit만을 실을 수 있고, STAP은 하나의

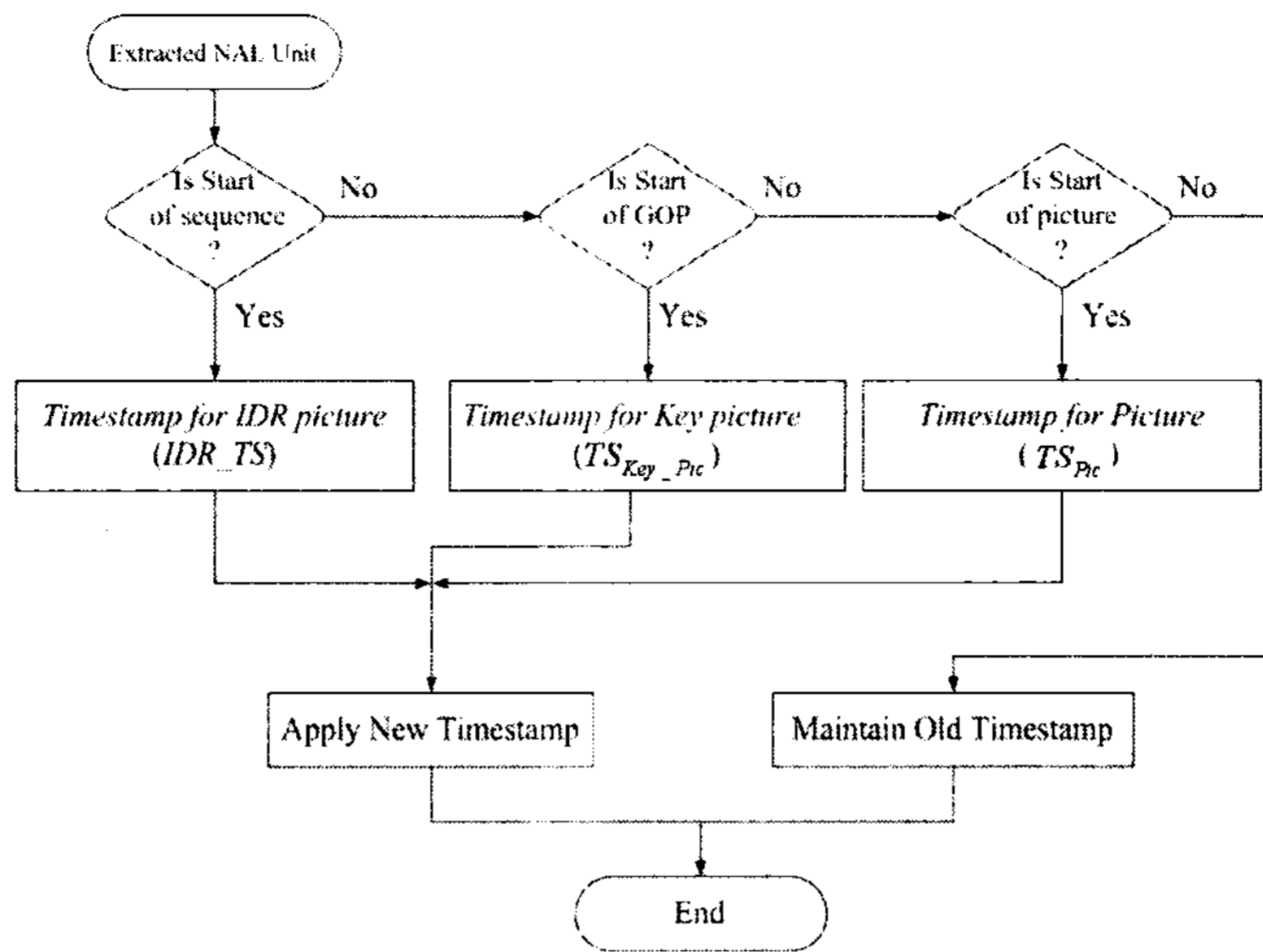


그림 8. NAL unit 단위로 RTP 타임스탬프 자동 생성 방법의 적용을 위한 흐름도

RTP 패킷에 동일한 화면 표현 시점에 속하는 다수의 NAL unit들을 동시에 실을 수 있다. 그리고 FU-A 타입은 하나의 NAL unit의 크기가 네트워크의 MTU (maximum transfer unit) 크기를 초과할 경우, 전송 도중 라우터나 게이트웨이에서 패킷분할을 방지하기 위하여 NAL unit을 MTU 크기를 초과하지 않도록 두개 이상으로 분할하여 각각의 분할된 단위를 개별적인 RTP 패킷에 나누어 실는 방식이다.

그림 9와 10은 제안된 방법을 실제 SVC 비디오 전송에 적용한 구현 결과를 보이는데, 그림 9는

SVC 비디오 송신단의 GUI 환경을 나타내고 그림 10은 수신단에서 수신한 RTP 패킷을 NAL unit 단위로 모으고 각 NAL unit별로 할당된 타임스탬프 값을 측정하여 표시한다.

그림 9의 오른쪽 윈도우에서는 주어진 채널 대역폭 (빨간색) 을 고려하여 실제 전송되는 데이터율을 나타내고, 가운데의 윈도우에는 현재 주어진 채널 대역폭 조건에서 전송 가능한 화면의 해상도, 화면율, FGS 데이터의 절단 (truncation) 위치 등을 3차원으로 표시한다. 왼쪽 윈도우에는 현재 전송되는 RTP 패킷의 패킷타입 (SNU, STAP-A, FU-A)과

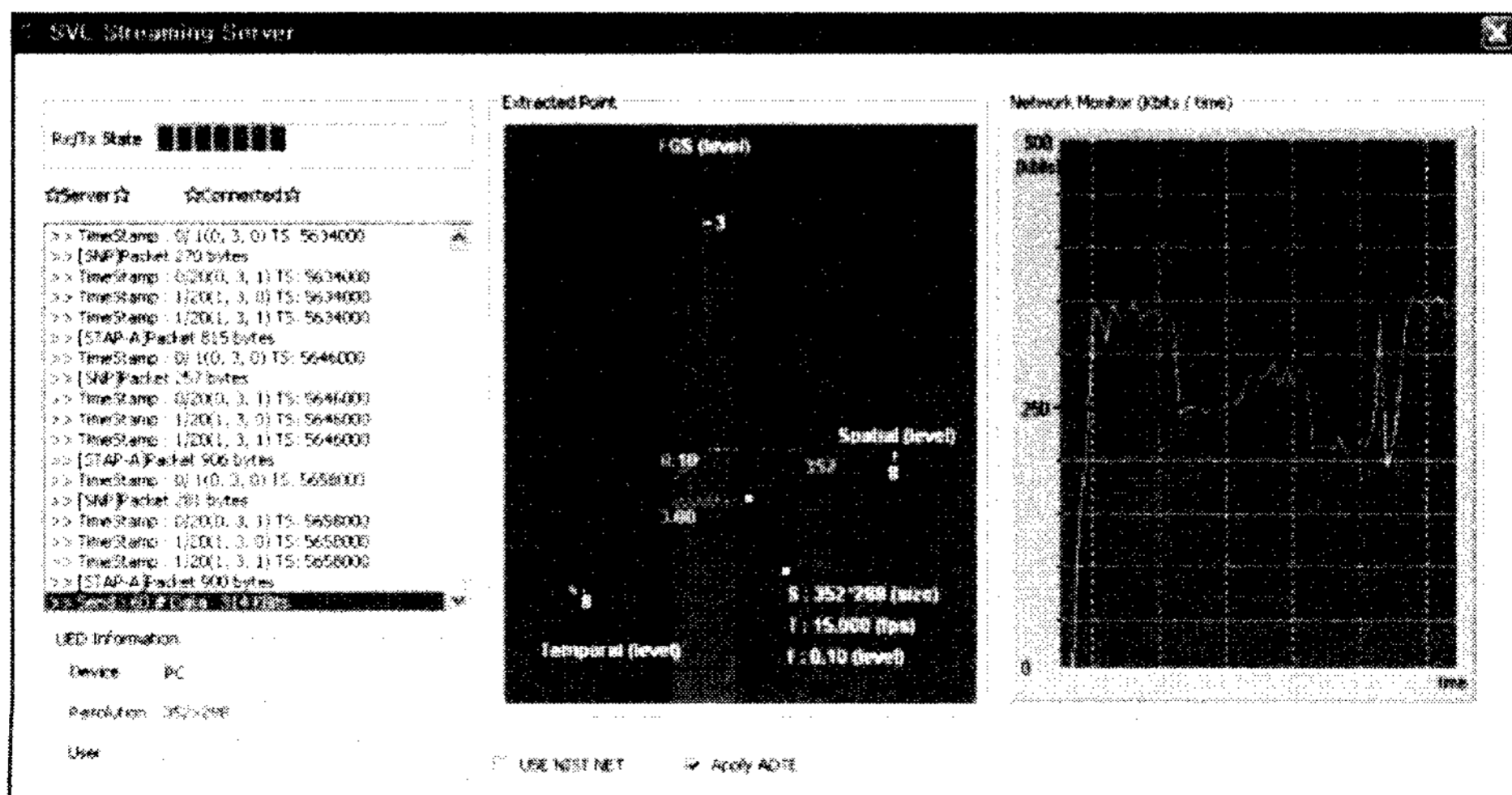


그림 9. SVC 비디오 송신단의 GUI 환경

NAL unit 헤더에 기록되는 NAL unit 타입과 (TID, DID, QID) 필드 값이 표시된다. 그림 10의 첫 번째 줄에 표시된 타임스탬프 (TS) 값 48000은 GOP의 첫 번째 화면인 키 픽처에 적용이 된 값이다. 각 화면마다 기본계층과 3개의 확장계층 정보에 해당하는 4 개의 NAL unit으로 구성되므로 각 화면마다 총 4개의 타임스탬프 값을 갖게 되며 동일한 화면에 속하는 4개의 NAL unit들은 그림 6과 같이 동일한 타임스탬프 값을 나타내고 있다. 키 픽처를 제외한 나머지 B-픽처에 대해서도 그림 8의 과정을 거쳐 자동적으로 타임스탬프 값이 할당되어 RTP 패킷 헤더에 기록된 후 전송이 된다. 그림 10에 표시된 타임스탬프 측정 결과는 그림 6에 표시된 타임스탬프 값과 일치하게 되어 제안된 방법의 정확성을 확인 및 검증할 수 있다.

```
>> Send : 1 # Data 135 KBits
>> TimeStamp : 3/ 1(0, 0, 0) TS: 48000
>> [FU-A]Packet 4936 bytes
>> TimeStamp : 3/20(0, 0, 1) TS: 48000
>> [FU-A]Packet 3547 bytes
>> TimeStamp : 3/20(1, 0, 0) TS: 48000
>> [FU-A]Packet 5177 bytes
>> TimeStamp : 3/20(1, 0, 1) TS: 48000
>> [FU-A]Packet 3702 bytes
>> TimeStamp : 2/ 1(0, 0, 0) TS: 24000
>> [SNP]Packet 571 bytes
>> TimeStamp : 2/20(0, 1, 1) TS: 24000
>> TimeStamp : 3/20(1, 1, 0) TS: 24000
>> TimeStamp : 3/20(1, 1, 1) TS: 24000
>> [STAP-A]Packet 1391 bytes
>> TimeStamp : 1/ 1(0, 0, 0) TS: 12000
>> [SNP]Packet 337 bytes
>> TimeStamp : 1/20(0, 2, 1) TS: 12000
>> TimeStamp : 2/20(1, 2, 0) TS: 12000
>> TimeStamp : 2/20(1, 2, 1) TS: 12000
>> [STAP-A]Packet 825 bytes
>> TimeStamp : 1/ 1(0, 0, 0) TS: 36000
>> [SNP]Packet 342 bytes
>> TimeStamp : 1/20(0, 2, 1) TS: 36000
>> TimeStamp : 2/20(1, 2, 0) TS: 36000
>> TimeStamp : 2/20(1, 2, 1) TS: 36000
>> [STAP-A]Packet 863 bytes
>> TimeStamp : 0/ 1(0, 3, 0) TS: 6000
>> [SNP]Packet 160 bytes
>> TimeStamp : 0/20(0, 3, 1) TS: 6000
>> TimeStamp : 1/20(1, 3, 0) TS: 6000
>> TimeStamp : 1/20(1, 3, 1) TS: 6000
>> [STAP-A]Packet 370 bytes
>> TimeStamp : 0/ 1(0, 3, 0) TS: 18000
>> [SNP]Packet 207 bytes
>> TimeStamp : 0/20(0, 3, 1) TS: 18000
>> TimeStamp : 1/20(1, 3, 0) TS: 18000
>> TimeStamp : 1/20(1, 3, 1) TS: 18000
```

그림 10. 수신단에서 측정된 각 NAL unit 별로 할당된 RTP 타임스탬프 값 (TS)

V. 결 론

본 논문에서는 SVC에 의해 압축된 비디오를 인터넷과 같은 IP망을 통해 전송하기 위해 RTP 패킷화할 때 RTP 패킷 헤더에 기록될 타임스탬프 값을 자동으로 생성하는 방법을 제안하였다. 제안된 방법은 SVC의 화면구조에서 동일한 TID 값을 갖는 화면들의 묶음인 TL_Group에 속하는 화면들의 TL_Group 내에서의 순서와 이 화면들이 갖는 타임스탬프 값의 상관관계를 관찰하여 얻은 일정한 규칙을 기반으로 개발되었다. 따라서, 제안된 방법에서는 NAL unit 헤더에 포함되는 TID 정보만을 사용하기 동작하기 때문에 동작구조가 간단하고 계산 복잡도 또한 낮은 장점이 있다. 제안된 방법은 SVC 비디오 전송 환경에서 RTP 패킷화에 필수적인 RTP 타임스탬프 값을 자동적으로 생성할 수 있기 때문에 오디오 스트림과의 입술 동기화가 요구되는 비디오 전송 서비스에 효과적으로 활용할 수 있다. SVC 비디오 압축 표준화가 마무리 되는 시점에서 압축된 SVC 비디오를 RTP패킷을 이용하여 전송하기 위해서 필수적인 기술적 문제점을 논리적으로 해결한 것은 매우 의의가 있다.

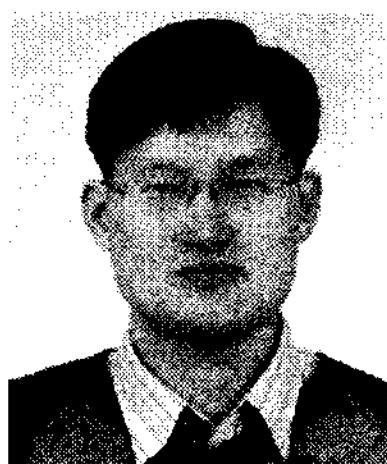
참 고 문 헌

- [1] D. Wu, Y. Hou, and Y. Zhang, "Transporting real-time video over the Internet: Challenges and approaches," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1855-1877, Dec. 2000.
- [2] Damien Stolarz, "Mastering Internet Video: A guide to streaming and on demand video," *Addison-Wesley*, Aug. 2004.
- [3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Real-time transport protocol," *IETF RFC 3550*, July 2003.
- [4] S. Wenger, M. Hannuksela, M. Westerlund, D. Singer, "RTP payload format for H.264 video," *IETF RFC 3984*, Feb. 2005.
- [5] S. Wenger, "RTP Payload Format for SVC Video," *IETF Internet Draft: draft-ietf-avt-rtp-svc-02.txt*, July 2007.
- [6] T. Wiegand, G. Sullivan, J. Reichel, M. Wien, "Joint draft 9 of SVC amendment," *Joint Video Team, JVT-V201*, Marrakech, Morocco, Jan. 2007.

- [7] H. Sun, A. Vetro, and J. Xin, "An overview of scalable video streaming," *Wireless Communications and Mobile Computing*, vol. 7, no. 2, pp. 159-172, Feb. 2007.
- [8] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable H.264/MPEG4-AVC Extension," in *Proc. IEEE International Conference on Image Processing*, pp. 161-164, Atlanta, USA, Oct. 2006.
- [9] D. Singer, M. Visharam, and Y. Wang, "Information Technology-Coding of audio-visual objects- Part 15: AVC file format, Amendment 2: File format support for scalable video coding," ISO/IEC 14496-15/FPDAM 2 (SVC file format), July 2007.

서 광 덕 (Kwang-deok Seo)

종신회원



1996년 2월 KAIST 전기 및 전자공학과 학사

1998년 2월 KAIST 전기 및 전자공학과 석사

2002년 8월 KAIST 전자전산학과 박사

2002년 8월~2005년 2월 LG전

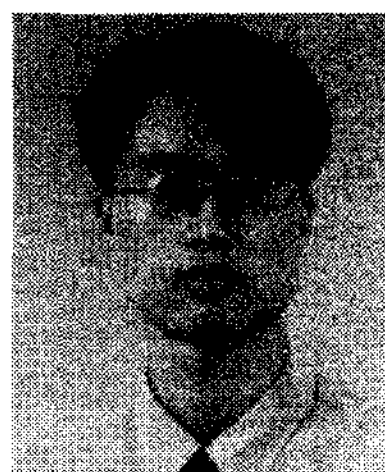
자 선임연구원

2005년 3월~현재 연세대학교 컴퓨터정보통신공학부 부교수

<관심분야> 영상부호화, 영상통신, 디지털 방송, 멀티미디어 통신시스템

정 순 흥 (Soon-heung Jung)

정회원



2001년 2월 부산대학교 전자공학과 학사

2003년 2월 KAIST 전자전산학과 석사

2005년 4월~현재 ETRI 방통미디어연구부 융합미디어연구팀 선임연구원

<관심분야> 영상부호화, 멀티미디어방송, 영상통신

김 재 곤 (Jae-gon Kim)

정회원



1990년 2월 경북대학교 전자공학과 학사

1992년 2월 KAIST 전기및전자공학과 석사

2005년 2월 KAIST 전자전산학과 박사

1992년~2007년 2월 ETRI 방송

미디어연구그룹 팀장

2007년 9월~현재 한국항공대학교 항공전자 및 정보통신공학부 조교수

<관심분야> 비디오 신호처리/코딩, 디지털방송 미디어, 미디어 컨버전스, 멀티미디어 통신

유 정 주 (JeongJu Yoo)

정회원



1982년 광운대학교 전자통신공학과 학사

1984년 광운대학교 전자통신공학과 석사

2001년 영국 LANCASTER 대학교 컴퓨터공학과 박사

1984년~현재 ETRI 방통미디어

연구부 융합미디어연구팀장

<관심분야> 영상통신, 비디오부호화, Scalable Video Coding, MPEG Video, ITU-T VCEG, 비디오 스트리밍, 스케일러블 스트리밍, IPTV