

# H.263과 G.723.1에 기반한 인터넷 화상회의 시스템

박 일환<sup>○</sup> 차 호정  
광운대학교 컴퓨터학과

## An Internet Video Conferencing System Based on H.263 and G.723.1

Ilhwan Park and Hojung Cha  
Department of Computer Science, Kwangwoon University

### 요 약

본 논문에서는 H.263과 G.723.1을 이용한 인터넷 화상회의 시스템의 구현을 기술한다. 시스템은 회의요청자의 요구에 대한 처리를 담당하는 프로세스와 미디어를 압축하여 전송하는 송신부, 그리고 상대방이 전송한 미디어를 복원하여 재생하는 수신부로 구성된다. 비디오와 오디오의 엔코더 및 디코더 모듈들은 소프트웨어 컴포넌트로 구현되어 다양한 스트리밍 시스템의 개발에 사용될 수 있다. 또한, IP 멀티캐스트, TCP/UDP 등의 프로토콜을 지원하여 다양한 형태의 회의를 가능하게 한다.

### 1 서론

통신기술의 발달과 멀티미디어 데이터의 처리속도 증가에 따라 다양한 멀티미디어 응용 프로그램이 출현하고 있다. 멀티미디어 응용프로그램의 전형적 예인 화상회의 시스템도 비약적인 발전을 이루고 있다. 서로 다른 장소에서 다양한 멀티미디어 데이터를 전송받아 회의를 진행하는 화상회의 프로그램의 경우, 컴퓨터 지원 공동작업의 핵심 기술로서 그 중요성이 증가되고 있다. 초기에 화상회의에 사용하기 위해 개발되었던 H.261 코덱은 계속 발전되어 H.263, H.263+에까지 이르렀다. 또한 오디오 압축분야에서도 낮은 대역폭을 필요로 하는 코덱의 연구를 통해 현재 인터넷에 적용이 가능한 G.723.1 오디오 코덱이 개발되었다.

현재 인터넷에서 사용가능한 다양한 화상회의 시스템이 개발되어 사용되고 있다. Mbone 상에서 동작하는 대표적인 화상회의 시스템으로 vic/vat[1]은 멀티캐스트 라우터를 가상의 터널로 연결하고, 이 터널을 통해 다수의 사용자간에 데이터를 멀티캐스트하여 회의를 진행한다. 코넬 대학에서 개발된 CU-SeeMe[2]는 인터넷 상에서 다수의 사용자가 서로 회의가 가능하게 구현된 화상회의 시스템으로 중앙집중형의 회의시스템을 가지고 있다. 마이크로 소프트사의 NetMeeting[3]은 중앙집중형 회의 시스템으로 회의 서버인 ULS를 통해 현재 인터넷에 연결되었는 사용자들간에 오디오와 비디오 및 텍스트등을 전송하여 회의를 진행한다. 특히 오디오의 전송에 있어서 전이중방식(Full Duplex)을 택하고 있어 사용자간의 오디오 송수신이 동시에 이루어진다. 이러한 다양한 인터넷 응용 화상회의 시스템은 낮은 대역폭의 인터넷의 특성상 비디오 전송시 낮은 프레임레이트의 서비스가 불가피하다.

본 논문에서는 저대역폭 인터넷을 대상으로 구현된 화상회의 시스템에 대해 기술한다. 화상회의 시스템은 소프트웨어로 구현된 H.263과 G.723.1 엔코더와 디코더로 구현되었다. 각 루틴은 DLL 형태의 컴포넌트로 구성되어 시스템 유지보수와 재사용성에 적합하다. 효율적인 버퍼관리기법을 통해 네트워크 혼잡에 대한 적절한 처리가 가능하고, 회의 참여자간의 시간차이를 최소화하는 기법을 제시하였다. 오디오 전송시에는 전이중방식을 사용하여 오디오 송수신이 동시에 이루어지도록 구현하였고 RTP를 사용하여 각각의 미디어에 대한 동기화 기능을 제공한다.

본 논문의 구성은 다음과 같다. 2절에서는 화상회의 시스템에 사용되는 H.263과 G.723.1 코덱에 대해 설명하고 이들간의 동기화에 사용되는 RTP에 대해 기술한다. 3절에서는 화상회의

시스템의 구성모듈과 프로토콜, 각각의 모듈에 사용된 API에 대해 설명하고, 시스템의 구현방법과 실험결과에 대해 기술한다. 4절에서 결론을 맺는다<sup>1</sup>.

### 2 인터넷 화상회의의 시스템 기술

논문에서 구현된 화상회의의 시스템은 비디오와 오디오 코덱으로 H.263[4]과 G.723.1[5]을 사용한다. 많은 움직임을 필요로 하지 않는 화상회의 특성상 이를 반영하는 효율적인 비디오 코덱이 개발되어 왔다. 인트라 프레임 압축후 다음 프레임의 변화량을 압축하는 H.261 코덱은 움직임이 많지 않은 화상회의에서 뛰어난 압축률을 제공한다. 논문에서 사용된 H.263은 이러한 H.261 코덱의 확장코덱으로 H.261에 비해 동일한 화질에서 상대적으로 적은 비트열을 발생시키기 때문에 대역폭을 절약할수 있는 잇점이 있다.

오디오 코덱으로 사용되는 G.723.1은 ITU-T에서 표준화한 알고리즘으로, 5.3kbps/6.3kbps의 이중레이트의 압축률을 지원한다. G.723.1 오디오 코덱도 회의의 특성상 대부분이 사람의 오디오라는 특성이 반영되었다. 따라서 음악등과같은 고주파 영역보다는 사람의 오디오에 해당하는 중,저주파 영역을 위주로 압축이 이루어져 높은 압축률을 제공한다. G.723.1은 저대역폭 화상회의의 프로토콜인 H.324의 오디오 부분에 해당되며 인터넷을 이용한 다양한 멀티미디어 서비스에 응용이 가능하다.

비디오와 오디오의 동기화에는 RTP[6]를 사용한다. RTP는 오디오와 비디오와 같은 실시간 미디어를 전송하기 위해 개발된 프로토콜로서 TCP/UDP와는 달리 사용자 어플리케이션 레벨에서 구현된다. 대표적인 화상회의의 시스템인 vic/vat에서 RTP를 사용한다. 또한 IP 멀티캐스트, TCP/UDP등의 다양한 전송 프로토콜을 지원함으로써 회의 상대방의 위치에 따라 적절한 전송 프로토콜을 선택하여 수행된다. IP 멀티캐스트의 경우에는 하나의 스트림으로서 다수의 사용자와 회의가 가능하므로 Mbone상의 다중회의기능이 가능해진다. 반면에 지역을 벗어나는 경우에는 터널링의 과정을 거쳐야하는 문제점이 있다. 따라서 TCP/UDP 프로토콜을 사용하여 원거리 사용자와 인터넷을 통해 회의가 가능하다.

### 3 시스템 구현

다음에는 화상회의의 시스템의 구성모듈과 프로토콜, 시스템 구성

<sup>1</sup>본 연구는 한국통신의 '99년도 정보통신기초 연구사업에 의해 지원받았음

에 사용된 API, 미디어 동기화 기법에 대해 기술한다.

화상회의 시스템 구성모듈 그림 1은 전체 시스템의 구성을 보여 준다. 송신부는 H.263 압축 모듈과 G.723.1 압축 모듈로 나누어지며, 수신부는 H.263 복원 모듈과 G.723.1 복원 모듈로 나누어진다. 또한 비디오와 오디오의 동기화를 위해 RTP를 사용하는 동기화 모듈이 있다.

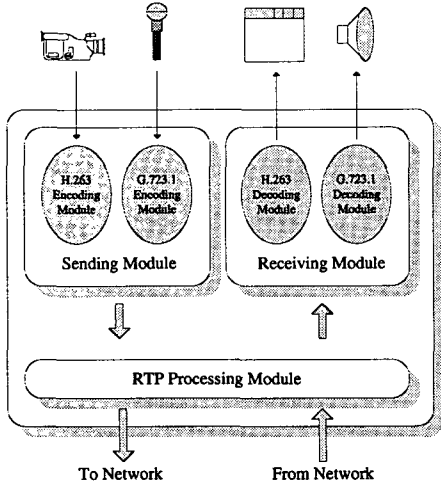


그림 1: 전체 시스템 모듈 구성

데이터 송신부의 H.263 압축 모듈에서는 실시간으로 입력받은 비디오 신호를 압축하고, G.723.1 압축 모듈에서는 마이크를 통해 입력받은 오디오 신호를 실시간으로 압축하여 상대방에게 전송한다. H.263 압축 모듈에서는 비디오 캡처와 압축사이의 시간차이에 의한 CPU사용시간 낭비를 방지하기 위해 캡처와 압축을 두개의 독립적인 쓰레드로 나누어 구성하였다[7]. H.263 엔코더는 ITU-T에서 제공하는 표준안에 따라 작성된 TMN 엔코더를 사용하여 구현되었다. G.723.1 압축 모듈은 실시간으로 사운드 데이터를 입력받아 압축하는 기법이 요구되는데, 기본적으로 윈도우즈에서 제공하는 API를 사용하는 경우 캡처시 시간지연이 발생하기 때문에 실시간으로 비디오의 캡처와 압축이 이루어지는 비디오와 동기화 문제가 발생한다. 따라서 G.723.1 압축 모듈에서는 마이크로소프트사의 DirectSound[8] API를 사용하여 캡처시의 시간간격을 최소화한다. 또한 오디오의 송수신을 동시에 가능하도록 전이중방식으로 구현하였다. 마이크를 통해 입력받은 PCM 포맷의 데이터를 한프레임에 해당하는 크기로 G.723.1 엔코더로 보내고 엔코더에서는 해당 프레임을 압축하여 G.723.1 스트림을 생성한다. 이렇게 생성된 G.723.1 패킷을 일정크기로 모아 하나의 RTP 패킷을 생성하여 상대방의 수신부로 전송한다. G.723.1 엔코더는 ITU-T에서 제공하는 표준안에 따라 작성된 코덱을 사용하여 구현되었다. H.263 엔코더와 G.723.1 엔코더는 API로 정의되어 DLL화 되어있다. 따라서 핵심이 되는 엔코더 루틴은 제공되는 DLL의 호출함으로써 Visual C++나 델파이등의 다양한 언어를 사용하여 화상회의의 송신부를 구현할수 있다.

데이터 수신부에서는 상대방이 전송한 비디오와 오디오 데이터를 받아 사용자에게 재생하여 회의를 진행한다. 비디오와 오디오의 복원모듈은 공통적으로 원형버퍼 구조로 구성되어있다. 원형 버퍼의 사용시 버퍼링을 하는 경우와 하지않는 경우로 나누어 동작이 가능한데, 버퍼링을 하지않는 경우 실시간으로 회의를 진행할수 있지만 네트워크의 혼잡이 발생하는 경우 비디오와 오디오 재생시 지터가 발생한다. 반면에 버퍼링을 하는 경우에

는 회의 참가자간에 미디어 재생시 시간차이가 발생하게 되지만 네트워크 혼잡에 능동적인 대처가 가능하다. 수신부의 H.263, G.723.1 디코더도 API로 정의되어 DLL화 되어있기 때문에, 다양한 언어로서 수신부 구현이 가능하다. 각각 독립적인 비디오와 오디오 코덱을 사용하기 때문에 이들 미디어간의 동기화 기능도 요구되는데, 수신부에서는 RTP를 사용하여 비디오와 오디오의 동기화 기능을 수행한다.

화상회의의 프로토콜 송신부와 수신부간에는 상대방의 회의요청에 대한 처리를 담당하는 디먼 프로세스가 동작한다. 디먼은 화상회의의 시스템과는 독립적인 프로세스로 백그라운드에서 실행되며 회의 요청자의 요구 메시지가 도달한 경우 동작한다. 회의 참여자간의 프로토콜 흐름도는 디먼 프로세스부, 회의 요청부, 요구 수락부, 미디어 초기화부로 구성된다. 참여자간의 프로토콜은 다음과 같은 순서로 이루어진다. 회의를 요청하는 사용자는 회의 요청부를 통해 상대방 IP 주소로 요청메시지를 보내고 상대방 시스템상에서 동작중인 디먼 프로세스는 해당 메시지를 시스템상의 요구 수락부로 전송한다. 디먼 프로세스에 의해 출력된 다이얼로그박스를 통해서 사용자는 회의요청을 허락할것인지 거절할것인지에 대한 여부를 요구 수락부로 전송한다. 사용자가 회의요청을 거부하거나 지정된 시간이 초과되도록 사용자의 응답이 없는 경우, 요구 수락부는 상대방에게 회의거부 메시지를 전송한다. 요청을 수락할 경우에는 미디어 초기화부로 제어가 넘겨져 해당 엔코더와 디코더에 대한 초기화를 수행하고 회의를 시작한다. 그림 2는 이러한 디먼 프로세스와 시스템간의 흐름을 나타낸다.

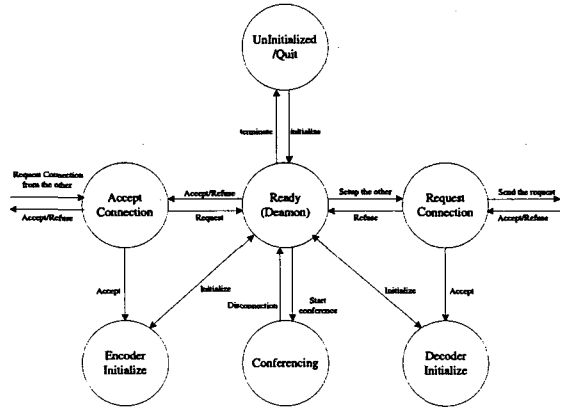


그림 2: 시스템 흐름도

API 구조 송신부와 수신부의 엔코더, 디코더 모듈은 DLL 형태로 구성된 API를 호출하여 구현되었다. 이러한 구조를 통해 소프트웨어에 대한 유지보수 및 재사용성을 증가시킨다. 다음은 각각의 API의 중요함수에 대한 간략한 설명을 나타낸다.

**H.263 Encoder**

- H263InitEncoder: 엔코더 초기화
- H263CloseEncoder: 엔코더 종료
- H263CreateCaptureWindow: 비디오 캡처윈도우를 생성
- H263SetEncoderParams: 각종 설정값을 설정
- H263SetEncodingMode: 압축모드를 설정
- H263StartEncoder: 엔코더 시작
- H263StopEncoder: 엔코더 중지
- H263SetEncoderNetworkParams: 네트워크 정보를 설정

**H.263 Decoder**

- H263InitDecoder: 디코더 초기화
- H263CloseDecoder: 디코더 종료
- H263CreateDecodingWindow: 디코더 윈도우를 생성
- H263SetDecoderParams: 디코더의 각종 설정값을 설정
- H263StartDecoder: 디코더를 시작
- H263StopDecoder: 디코더를 중지
- H263PauseDecoder: 디코더를 순간정지
- H263SetNetworkParams: 네트워크 정보를 설정

**G.723.1 Encoder**

- G723InitEncoder: 엔코더를 초기화
- G723CloseEncoder: 엔코더를 종료
- G723StartEncoding: 엔코더를 시작
- G723StopEncoding: 엔코더를 중지
- G723SetEncoderParams: 엔코더의 각종 설정값을 설정
- G723SetEncoderRate: 압축레이트 설정 (5.3/6.3kbps)
- G723SetEncoderNetworkParams: 네트워크 정보를 설정

**G.723.1 Decoder**

- G723InitDecoder: 디코더를 초기화
- G723CloseDecoder: 디코더를 종료
- G723DecoderStart: 디코더를 시작
- G723DecoderStop: 디코더를 중지
- G723SetDecoderParams: 디코더의 각종 설정값을 설정
- G723SetDecoderNetworkParams: 네트워크 정보를 설정

**RTP를 사용한 미디어 동기화** 화상회의에는 각각 독립적인 비디오와 오디오 코덱이 사용되므로 이들간의 동기화 기능이 필요하다. 단순히 비디오와 오디오를 전송받은 순서대로 재생을 하게 되면 네트워크의 혼잡등의 이유로 원하는 데이터가 제시간에 도착하지 못하는 경우 비디오나 오디오가 끊어지는 문제가 발생한다. 따라서 본 논문의 시스템에서는 RTP의 타임스탬프 필드를 사용하여 동기화를 제공하는 방법을 사용한다. RTP의 타임스탬프 필드는 해당 패킷이 샘플링되어 생성된 시간이 들어가기 때문에 동일한 타임스탬프 필드를 갖는 패킷들을 재생하는 방법으로 서로간의 동기화를 이룬다. 송신부의 압축 모듈에서는 각각의 압축된 패킷에 현재 시간을 기록하여 상대방의 수신부로 전송을 하고 수신부의 복원 모듈에서는 전송받은 패킷의 RTP 정보를 분석하여 동일한 TS값을 가지는 패킷을 각각 복원하여 비디오와 오디오의 재생속도를 조절한다. 미디어 복원시는 오디오의 복원을 우선적으로 수행하고 해당 오디오 패킷의 타임스탬프와 동일한 타임스탬프값을 가지는 비디오 패킷을 선택하여 복원하는 방식으로 수행된다.

**화상회의 시스템의 실행결과** 그림 3은 화상회의 시스템의 실행모습이다. 시스템의 송신부측 윈도우에서는 카메라로부터 입력받은 비디오 화면을 출력하고 실시간으로 비디오를 압축하여 회의 상대방에게 전송한다. 오디오 역시 마이크를 통해 입력받은 오디오 신호를 실시간으로 압축하여 비디오와 동시에 상대방에게 전송한다. 수신부측 윈도우에서는 송신부에서 전송한 비디오와 오디오 패킷을 받아 복원하여 재생한다. 송신부측의 비디오 압축시 다양한 압축 옵션을 통해 압축되는 비디오의 화질 및 요구 대역폭을 사용자가 조절할수있는 기능을 제공한다. 동작상황의 모니터링을 위해 송신부측에서는 압축되는 비디오와 오디오의 각종정보 및 비디오 프레임레이트, 비트레이트등의 정보를 제공하고, 수신부측에서는 비디오 프레임레이트, 평균 비디오 프레임레이트, 비디오 비트레이트, 평균 비디오 비트레이트, 오디오 비트레이트, 현재 버퍼의 비디오와 오디오 데이터의 양을 출력한다. 또한 비트레이트와 프레임레이트에 대한 시각적인 정보제공을 위해 모니터링 윈도우를 통해 시간흐름에 따른 정보의 변화를

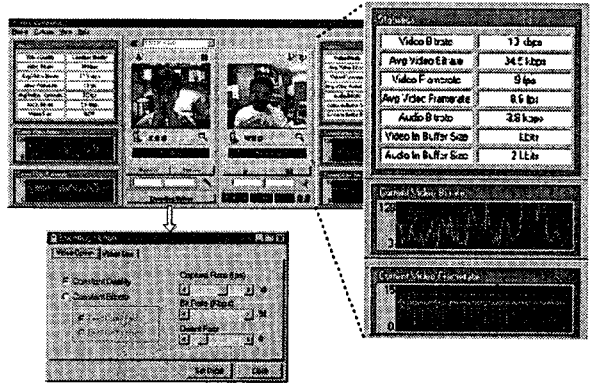


그림 3: 시스템 실행모습

출력한다.

**4 결론 및 향후과제**

본 논문에서는 H.263과 G.723.1에 기반한 인터넷 화상회의 시스템의 구현에 대해 기술하였다. 시스템은 상대방의 회의요청에 대한 처리를 담당하는 디번 프로세스와 송신부, 수신부 모듈로 구성되고, 각각의 독립적인 미디어의 동기화를 위한 RTP 모듈이 포함된다. 독립적으로 실행되는 디번 프로세스를 통해 상대방에 대한 회의요청에 대해 선택적으로 회의를 시작한다. 화상회의 시스템은 소프트웨어로 구현된 H.263과 G.723.1 엔코더와 디코더로 구현되었다. 각 루틴은 DLL 형태의 콤포넌트로 구성되어 시스템 유지보수와 재사용성에 적합하다. 각종 설정값등을 통해 압축되는 상태를 조절할수있고 다양한 통계정보를 통해 현재 동작되는 상태를 모니터링할수 있다. 송신부와 수신부 모듈은 DLL 형태로 제공되는 API를 호출하여 델파이4.0을 이용하여 구현되었다.

향후 다자간 회의기능과, RTCP를 사용한 피드백 정보를 통해 자동으로 현재 네트워크 상태에 최적으로 대역폭을 조절하는 기법에 대한 연구가 진행중이다.

**참고 문헌**

- [1] vic/vat, <http://www.nrg.ee.lbl.gov/vic/>.
- [2] CU-SeeMe, <http://ask.simplenet.com/cu/cuhelp.htm>.
- [3] Microsoft NetMeeting, <http://www.microsoft.com/windows/netmeeting/>.
- [4] ITU-T Recommendation H.263, July 1995.
- [5] ITU-T Recommendation G.723.1, March 1996.
- [6] H.Schulzrinne, S.Casner, R.Fredrick, V.Jacobson, *RTP:A Transport Protocol for Real-Time Application*, RFC 1889, February 1996.
- [7] 박일환, 차호경, 'H.263/G.723.1을 이용한 인터넷 스트리밍 시스템 구현' 한국정보과학회 '99 춘계 학술발표논문집, 1999년 4월.
- [8] Bradley Bargaen and Peter Donnelly, *Inside DirectX*, Microsoft Press, 1998.