

# 음성/영상 스트리밍(streaming) 기술 연구동향

나승구\*, 안 종 석\*\*

\*동국대 박사과정, \*\*동국대 조교수

## 1. 서 론

최근 컴퓨터와 통신기술이 발달함에 따라 음성 및 영상 등의 멀티미디어 데이터를 네트워크를 통해 실시간적으로 전송할 수 있는 서비스에 대한 요구가 증가하고 있고, 이러한 요구를 만족시킬 수 있는 멀티미디어 처리기술과 전송기술의 연구가 활발하게 진행되고 있다. 즉 기존의 인터넷을 통한 FTP, Telnet, e-mail과 같은 텍스트 위주이며 비실시간적인 서비스 형태로부터 인터넷 라디오, 인터넷 TV와 같은 멀티미디어 데이터의 실시간 전송을 요구하는 다양한 서비스들이 등장하기 시작하였다.

이러한 새로운 서비스를 구현하기 위한 기술들은 기존의 텍스트 위주의 정보를 처리/전송하는 기술로는 만족할 수 없는 다음의 두 가지 특성을 가지고 있다. 첫째는 대용량의 데이터를 실시간적으로 처리할 수 있어야 하며, 둘째는 이러한 데이터를 네트워크 상황에 관계없이 응용 프로그램이 요구하는 정도로 실시간적이고 또한 안정적으로 전송할 수 있어야 한다. 특히 두 번째 요구조건은 현재의 네트워크 대역폭을 공유하는 인터넷과 같은 비연결성(connectionless) 패킷 스위칭 네트워크에서는 만족하기 어려운 것이다. 즉 인터넷은 네트워크를 다수의 사용자가 공유하는데 비해, 인터넷은 이들 사용자들의 데이터를 차별적으로 서비스하지 않는 best-effort 서비스만을 제공하고 있다. 또한 네트워크 자체에서 데이터의 흐름을 일정 통로로만 전송할 수 있는 연결성 서비스를 제공하지 않고 있다. 이러한 문제는 응용 프로그램에게는 데이터 전송 지연, 지연 지터(jitter), 그리고 데이터 손실이라는 현상으로 나타난다.

이러한 응용 프로그램의 요구 조건들을 만족하지 않는 네트워크 상황에서 서비스의 품질을 최대한 향상시키려는 기술들을 스트리밍 기술이라 통칭할 수 있는데, 이 스트리밍 기술은 크게 압축기술, 전송기술, 그리고 재생기술 세 분야로 나누어 볼 수 있다. 압축 기술은 멀티미디어의 대용량 데이터를 정보의 손실이 없이 데

이터 용량을 줄이면서 또한 패킷 손실이 발생할 경우에도 서비스의 품질을 저하시키지 않으려는 것이며, 전송 기술은 연결성 서비스를 중단에서 제공할 수 있게 하며, 서비스간에 동기화 정보를 제공하며, 대역폭이 서로 다른 네트워크에서도 품질에 저하되는 것을 방지하는 기술 등을 포함한다. 마지막으로 재생기술은 패킷 손실이 발생했을 때 복구 방식, 지연을 만회하는 기술들을 일컫는다. 이들 세 분야의 기술들은 서로 밀접하게 관련되어 있어 하나의 기술이 선택되면 거기에 알맞은 기법들이 정해지게 된다.

아직까지는 이들 스트리밍에 관한 연구는 진행되고 있는 단계로, 다양한 멀티미디어 응용 프로그램의 요구 조건들을 모두 수용하기에는 부족한 실정이다. 본 고에서는 최근에 소개되고 있는 음성/영상 스트리밍 기술과 연구동향을 소개하는 것으로, 2장에서는 스트리밍의 정의와 동작을 자세히 설명하고, 3장에서는 멀티미디어를 전송하기 위한 전송 프로토콜의 종류와 동작에 대해 살펴본다. 마지막 4장에서는 QoS(Quality-of-Service) 제공을 위한 기술에 대해 살펴본다.

## 2. 스트리밍 기술 개요

스트리밍<sup>[2]</sup>이란 마치 물이 흐르는 것과 같이 끊임없고 연속적인 흐름으로 데이터를 처리 할 수 있는 데이터 전송기술을 의미한다. 현재 스트리밍 기술은 인터넷의 폭발적인 성장과 더불어 on-line 응용 프로그램을 위한 중요한 기술로 부각되고 있다. 즉 네트워크의 대역폭이 증가됨에 따라 상호작용(interactivity)이 요구되고 따라서 스트리밍 기술이 중요시 되고 있다. 예를 들면 인터넷상에서 MPEG 압축 비디오 전체를 한번에 전송받아 상영하기 위해서는 장시간 기다려야 하는 불편함이 따르게 된다. 그러나 스트리밍 기술을 이용하게 되면 이들 파일이 모두 전송되기를 기다릴 필요 없이 데이터의 전송과 재생이 거의 동시에 이루어지도록 할 수 있다.

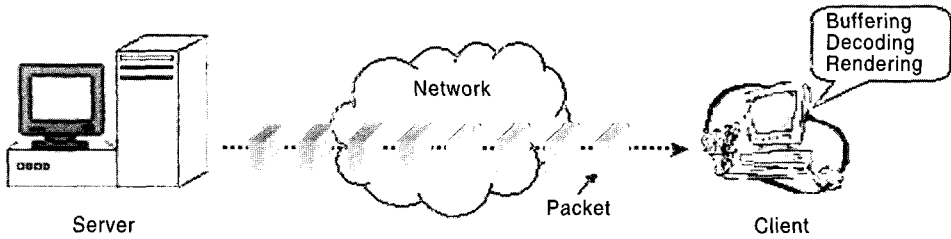


그림 1. streaming 개념도

스트리밍이 동작하기 위해서는, 송신자는 멀티미디어 소스(source)에서 생성되는 연속적인 데이터의 전송을 위해 패킷화(packetization)하여야 하며, 수신자측에서는 패킷에 담겨진 데이터를 모아서 다시 소스에서 생성된 것처럼 하나의 데이터 스트림으로 만들어 사운드나 그림으로 변환해주는 응용프로그램에 보내주어야 한다. 이때 송신자는 네트워크 전송 속도와 수신자의 수신 상태에 따라 알맞게 전송 데이터 양을 조정해야 하며 수신자는 여분의 버퍼를 이용하여 패킷의 지연을 보완하여 상영 시간에 알맞게 데이터를 공급해야 한다. 이때 전송 속도에 따른 불일치로 인한 패킷의 과도한 손실과 패킷의 지터는 데이터의 흐름을 방해하여 상영이 매끄럽지 못하게 된다. 그림 1은 클라이언트가 서버로부터 미디어 스트림을 네트워크를 통하여 전송 받는 과정과 처리하는 과정을 개념적으로 나타낸 것이다.

최근에는 많은 스트리밍 기술들이 경쟁적으로 출현되고 있는데 이들 스트리밍 기술을 이용하여 구현한 응용프로그램으로는 오디오 부문에서는 Audioactive, AudioSoft 등이 있고 오디오/비디오 부문에서는 RealAudio, Stream Works, VDOLive, Windows Media Player 등이 있다.

### 3. 멀티미디어 전송 프로토콜

#### 3.1 개요

현재 인터넷은 어떠한 보장없이 최대한 데이터를 빠르게 전송하는 best-effort 서비스를 제공하는 IP 프로토콜을 사용하므로, 전송지연, 손실률, 지터로 표현되는 사용자의 요구 또는 QoS를 완전히 충족시키지 못한다. 지터는 수신단의 버퍼를 이용하여 어느 정도는 해소할 수 있으나 패킷 손실은 영상이나 음성의 품질을 현저히 저하시킨다. 기존의 안정적 전송을 위한 트랜스포트 계층 프로토콜로는 TCP 프로토콜이 존재하나 TCP는 분실된 패킷에 대해 재전송을 하므로 재

전송으로 인한 왕복시간이라는 과도한 패킷 지연이 초래된다. 상영시간 이후에 도착한 패킷은 실시간 응용 프로그램에서는 패킷 손실로 간주되기 때문에 역시 멀티미디어 서비스의 품질을 크게 저하시키게 된다. 따라서 실시간성을 요구하는 응용프로그램에서는 수행속도가 빠르고 패킷 손실을 복구하지 않는 UDP/IP를 기반으로 멀티미디어 서비스가 인터넷상에서 제공되기도 한다.

그러나 UDP/IP 또한 멀티미디어 응용프로그램의 QoS 요구를 보장하지 않기 때문에 인터넷에서는 사용자의 요구만큼 서비스의 품질을 향상시키기는 어렵다. QoS를 보장하기 위해서는 수신단까지의 경로에 위치한 모든 라우터 또는 스위치에서 대역폭을 예약해야 하며, 예약된 대역폭에 따른 차별화된 서비스를 모든 라우터 또는 스위치들이 제공해야 하는 데 이러한 메커니즘이 인터넷 전부에 구현되기는 가까운 장래에는 불가능하리라 사료된다. 따라서 멀티미디어 서비스를 위한 현재의 스트리밍 기술은 기존 IP에 기반한 인터넷 프로토콜 구조를 변경하지 않고 단말에서 품질을 최대한 향상시키려는 방향으로 진행되고 있다.

#### 3.2 멀티미디어 관련 프로토콜

본 절에서는 인터넷을 기반으로 하는 스트리밍 프로토콜들을 소개한다. 송신단에서 패킷화에 사용되는 프로토콜, 자원 예약을 위한 RSVP(Reservation Protocol)<sup>[13]</sup>, 실시간 데이터 전송을 위한 RTP/RTCP(Real-time Transport Protocol/RTP Control Protocol)<sup>[11]</sup>, 세션과 관련된 SAP/SDP/SIP(Session Advertisement/Description/Invitation Protocol)<sup>[1]</sup> 그리고 실시간 스트리밍 처리를 위한 RTSP(Real-time Streaming Protocol)<sup>[12]</sup> 등을 소개한다. 이들 스트리밍에 관련된 멀티미디어 데이터 전송을 위해 제안된 프로토콜<sup>[1][15]</sup>간의 관계는 그림 2에 표현되어 있다.

##### 3.2.1 패킷화에 사용되는 프로토콜

인터넷은 패킷전송 방식이기 때문에 비디오 또는

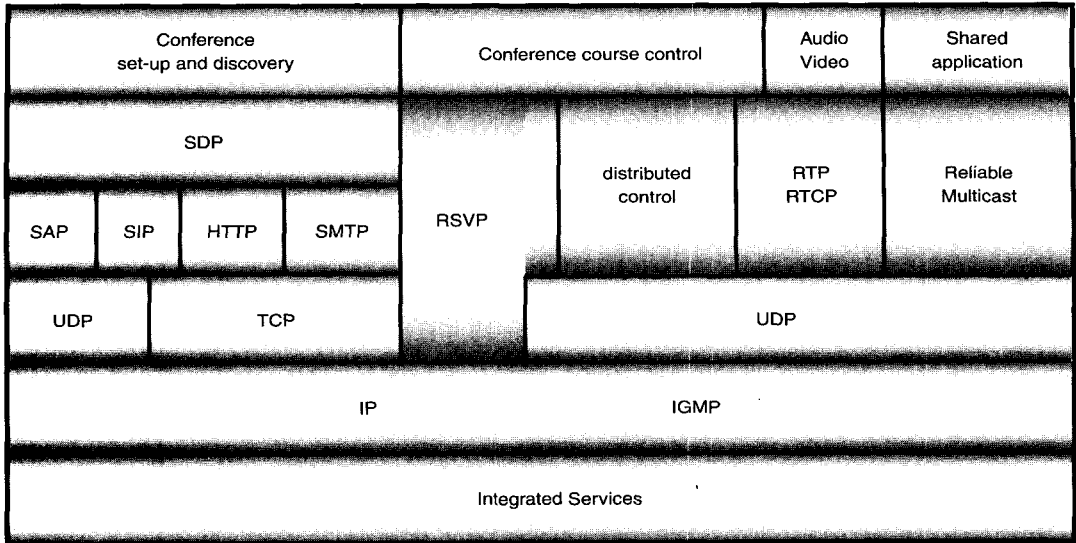


그림 2. 멀티미디어 관련 프로토콜 스택

오디오 스트림을 전송하기 위해서는 데이터를 전송하기 이전에 디지털화와 압축과정을 거쳐서 패킷이라 불리는 형태의 데이터로 분할하여야 하는데 이를 패킷화라고 한다. 우선 비디오나 오디오의 아날로그 신호를 디지털화 하기 위해서는 일정한 간격으로 샘플링(sampling)을 해야 한다. 이러한 샘플링 간격은 아날로그 신호의 대역폭에 따라 결정된다. 샘플링 된 신호를 양자화 과정을 거친 후 디지털 신호로 만들면 원래의 신호보다 훨씬 넓은 대역폭을 갖거나 또는 데이터 양이 증가한 디지털 신호가 된다.(예를 들어 음성 신호 대역폭은 300 ~ 3400Hz이지만 PCM 신호인 디지털화 된 음성신호는 최소 64kHz이상의 대역폭을 가진다.)

동영상과 같은 데이터는 양이 매우 많기 때문에 디지털화 된 신호를 작은 양으로 압축할 필요가 있다. 영상의 경우에는 H261, H263, MPEG 등의 알고리즘을 사용하고 음성의 경우에는 PCM, GSM, G.722 등의 알

고리즘을 사용한다. 압축과정을 통하여 생성된 데이터는 연속적인 대용량의 데이터이기 때문에 이를 네트워크 전송로의 최대전송단위(MTU)의 크기에 벗어나지 않도록 적당한 크기로 분할하여 전송하여야 한다. 이때 전송한 패킷에는 수신단에서 이전에 전송한 패킷이 분실되었을 경우에도 원래의 스트림으로 복원될 수 있도록 매 패킷마다 스트림의 복원에 필요한 정보를 헤더에 첨가하게 된다. 본 절에서는 패킷화 프로토콜의 예로써, H261과 MPEG2의 오디오 데이터의 압축방법과 결부한 패킷화 헤더를 설명한다.

그림 3은 RTP/RTCP를 통해 전송되는 H261<sup>[7]</sup>의 패킷 구조를 나타내고 있다. 헤더 부분에는 개별적 패킷을 상영할 수 있게 하는 정보들, 즉 GOB블록의 개수, 양자화(quantizer) 값, 수직/수평 움직임 벡터(motion vector)등이 포함되어 있다. 헤더 이후의 데이터 영역에는 Picture, GOB, MB, Block의 4계층으로 이루어진 H261 영상 데이터가 포함되어 있다.

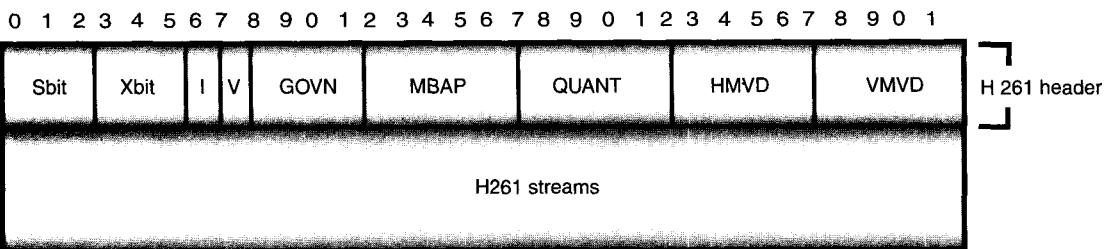


그림 3. H.261 패킷구조

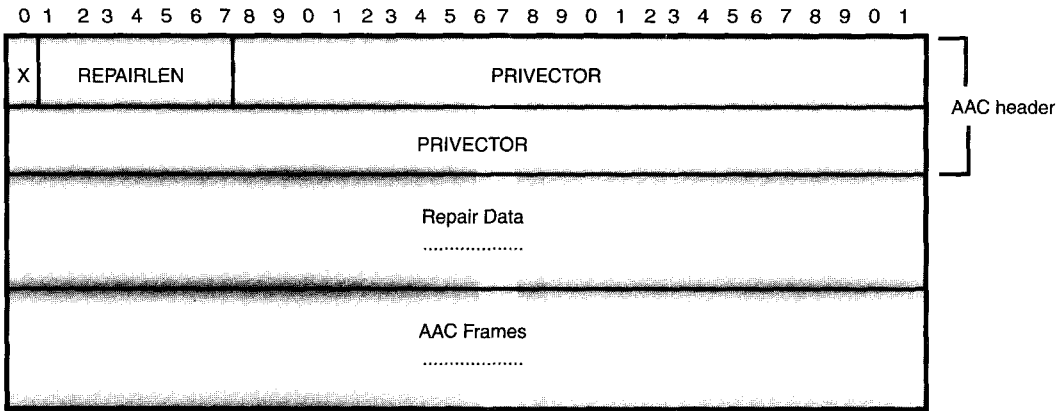


그림 4. AAC 패킷구조

그림 4는 음성 압축에 이용되는 ISO/IEC MPEG-2 AAC(Advanced Audio Coding)<sup>[18]</sup>의 데이터 패킷 구조를 보여주고 있다. 패킷은 크게 세 부분으로 구성되는데 AAC 헤더부분, 바로 전에 전송됐던 패킷의 데이터가 저장되어 있는 Repair Data부분, 현재의 음성 데이터를 포함하는 AAC Frames부분으로 구성된다. 헤더의 각 PRIVECTOR는 해당되는 Repair Data에 담겨있는 중복 데이터 품질의 정도를 표시한다. 즉, 바로 이전의 패킷이 손실되었을 때 Repair Data에 담겨있는 중복 데이터를 이용할 수가 있는 데, 이때 수신단의 상태에 따라 고품질의 그러나 데이터 양이 많은 Repair Data를 또는 저품질의 그러나 데이터 양이 적은 Repair Data부분을 사용할 수 있다. 마지막으로 Repair Data와 AAC Frames 부분에는 각각 해당되는 데이터와 더불어 일련번호와 데이터의 길이를 나타내는 정보들이 포함되어 있다.

프로그램들이 개발되어 사용되고 있지만 best-effort 서비스인 인터넷에서 멀티미디어 응용 서비스를 처리하기에는 부족함이 많다. IETF에서는 IntServ(Integrated Service)라는 그룹을 두어서 인터넷 서비스의 약점을 보완하기 위해 라우터에서 QoS를 제공하기 위한 기준 모델을 정의하였다<sup>[19]</sup>. 이 모델에 의하면 각 라우터에서 자원을 예약할 수 있는 모듈과 관리 모듈, 호 수락 제어 모듈에 대해 정의 하고 있으며, 자원 예약 프로토콜을 사용하여 정책을 수립한 후 이 정책에 따라서 라우터에 유입된 패킷들을 분류하고 플로우별로 각각 다른 큐를 제공함으로써 자원의 할당을 보장한다.

RSVP<sup>[20]</sup>에서 자원을 예약하는 과정은 다음과 같다. 자원을 예약하기 위해 송신자는 Path 메시지를 RSVP 수신 목적지로 전송한다. 이 Path메시지는 차후에 송신자로부터 또는 수신자로부터 오는 데이터가 같은 경로를 사용하기 위해, 즉 경로 설정을 하기 위해 사용된다. 참고로, 인터넷에서는 같은 송신자와 수신자일 경우에도 데이터는 수시로 다른 경로로 전송된다. RSVP 라우

3.2.2 자원 예약 프로토콜(RSVP)

영상전화, 원격진료, VOD 등 많은 멀티미디어 응용

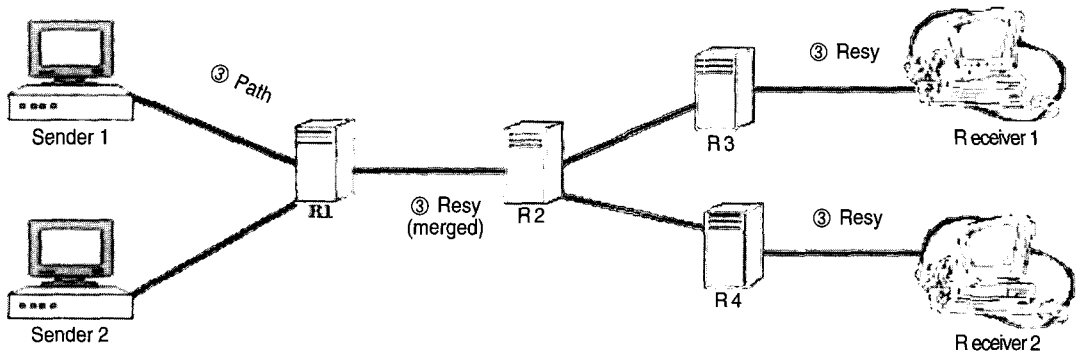
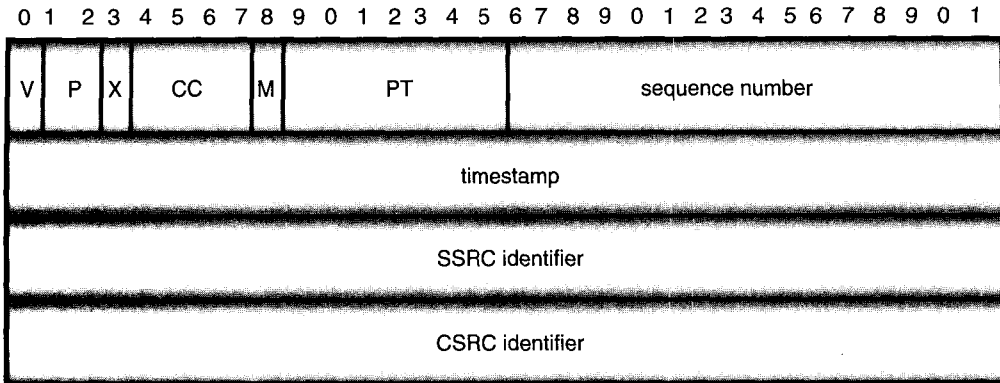


그림 5. RSVP 자원 예약 과정



터들은 Path메시지를 수신하면 Path 상태 정보를 유지하기 위해 자신의 테이블의 엔트리를 갱신한 후, Path 메시지를 생성하여 수신자 쪽으로 전송한다. Path 메시지가 수신되면 RSVP 수신자는 자원 예약을 요구하는 Resv 메시지를 Path메시지가 전달된 경로를 통해 송신자에게 다시 전송한다. Resv 메시지가 성공적으로 수신되면 송신자는 수신자에게 데이터 전송을 시작한다.

그러나 완벽한 자원 예약을 위해서는 RSVP를 비롯한 효율적인 스케줄러의 구현, 사용한 자원을 표현하는 문제, 자원 관리 등 여러 가지 난제가 있다. 또한 RSVP는 패킷을 flow(송신자와 수신자의 쌍으로 정의할 수 있음)별로 분류 관리해야 하기 때문에 flow의 숫자가 많아질 경우 스위칭 성능이 저하되는 확장성이 문제가 되고 있다. 이를 해결하기 위해 최근에 IETF에서는 flow별이 아닌 class로 관리하는 DS(Differentiated Service)[5][15]가 제안되고 있다.

3.2.4 실시간 전송 프로토콜(RTP)

RTP<sup>[11]</sup>는 영상 또는 음성과 같은 실시간 특성을 갖는 데이터를 전송하기에 적합하도록 IETF (Internet Engineering Task Force)의 AVT (Audio-Video Transport) 그룹에서 제안하여 RFC1889로 표준화된 프로토콜이다. RTP는 데이터를 전송하는 RTP와 데이터 전송을 제어하는 RTCP로 나뉜다. RTP는 오디오, 비디오 또는 시뮬레이션 데이터와 같은 실시간 데이터를 전송하기 위한 프로토콜이다. 그러나 RTP는 RSVP와 같이 자원을 예약하지 않으므로 사용자가 요구하는 QoS를 보장해 주지 않는다.

RTP는 주로 UDP 위에서 구성되는데, 이것은 UDP가 제공하는 멀티플렉싱과 체크섬의 기능을 이용하기 위해서다. 그러나 RTP가 트랜스 포트계층과는 독립적으로 설계되었기 때문에 반드시 UDP위에서 구성되어

야 하는 것은 아니다. RTP 자체로는 실시간 데이터 전송과 QoS에 대한 어떠한 보장도 해 주지 않는다. 또한 손실 없는 데이터 전송도 보장해 주지 않으며, 수신자에게 데이터가 순차적으로 전달될 것도 보장해 주지 않는다. 그림 6과 같이 단지 RTP는 타임스탬프(timestamp), 일련 번호(sequence number)등 실시간과 순서에 관계되는 정보만을 전달해 주고, 송수신 상태에 대해서는 RTCP 보고를 받을 뿐이다. 다음은 RTP와 RTCP에 대한 간략한 설명이다.

가. RTP

TCP와 같이 코드량이 크고 수행시간이 긴 프로토콜에 비해 RTP는 프로토콜이 간단하여 수행시간이 적기 때문에 실시간 응용프로그램에 알맞다. RTP는 실시간 응용프로그램을 위해 두 가지 메커니즘을 제공한다. 첫째, 일련번호를 제공하여 데이터를 순차적으로 처리할 수 있게 한다. 둘째, 타임스탬프 필드에 데이터가 생성된 시각 정보를 제공함으로써 데이터가 처리되어야 하는 시각(playback time)을 알려주어, 다른 미디어에서 생성된 데이터와 같이 처리될 때는 다른 미디어와 동기화(synchronization)를 할 수 있게 한다. 그림 6은 RTP 버전 2의 패킷 헤더 구조<sup>[11]</sup>를 나타낸다.

나. RTCP

RTCP는 제어 프로토콜로써 회의 또는 세션(Session)에 참가한 모든 참가자에게 피드백 정보를 주기적으로 전송한다. RTCP 프로토콜은 데이터를 전송하는 RTP와는 다르게 세션에 관계된 제어 패킷을 송수신하는데 그 목적이 있다. RTCP 제어 메시지를 이용할 경우 세션에 관한 정보는 물론이고 에러 복구와 흐름제어까지도 할 수 있다. RTCP는 다양한 제어 정보를 전송하는데 SR(Sender Report), RR(Receiver Report), SDES

(Source DEscription), BYE(goodBYE), APP(APPLIcation defined)의 5가지 메시지 형식을 가지고 있다. 기본적으로 SR과 RR을 이용하여 송신자와 수신자의 전송 상태를 파악할 수 있게 한다. SDES는 추가적인 정보를 전송할 때, BYE는 자신이 세션을 떠난다는 것을 다른 참여자에게 알릴 때, 그리고 APP는 응용 프로그램에서 새로운 종류의 메시지들을 이용한 실험과 테스트를 위해 사용한다.

### 3.2.5 세션 광고/기술/초청 프로토콜(SAP/SDP/SIP)

SAP/SDP/SIP[1]은 생성된 원거리 멀티미디어 세션에 참가하기를 원하는 참가자들을 알려주기 위해 사용되는 프로토콜들이다. SDP는 생성된 세션을 표현하기 위한 일종의 표현 언어이며, SAP는 SDP로 표현된 세션 정보를 전송하기 위한 프로토콜이다. 즉 SDP와 SAP관계는 HTML과 HTTP 관계와 같다. SAP는 세션 정보를 IP의 멀티캐스트 주소로 인터넷에 브로드캐스트하게 되면 세션에 참여하고자 하는 참가자는 그 광고 내용을 읽고 SDP에 의해 지정된 세션 주소를 사용하여 원하는 세션 가입할 수 있다.

구체적으로 SDP는 세션의 멀티캐스트 주소, 미디어 양식, 시간, WWW 주소 등을 기술한다. SDP의 초기 버전은 세션을 시작하고 회의의 할당을 표현하는 기능은 제공하였으나, 회의를 관리하기 위한 내용과 세션 광고시에 사용할 대역폭 정보, 사용 시간, 회의에 대한 설명을 기술하는 기능은 제공하지 않았다. 이에 비해 SDP 버전 2는 WWW 주소등을 포함한 세션 개설자의 정확한 정보, 사용 미디어 정보, 언어 문제, 대역폭 기술, 암호화 정보 등의 많은 내용이 추가되었다. SDP와 SAP를 구현한 예로는 LBL(Lawrence Berkeley Laboratory)의 SDR이 있다.

SIP는 세션 개설자가 특정 사용자를 초청할 때 사용하는 프로토콜로서 다양한 참가자들을 초청할 수 있다. 즉 진행중인 회의 내용을 기록하기 위한 미디어 저장장치나 VOD서버, 또한 이동 장비(mobile device)들도 위치에 상관없이 세션에 참여하도록 요청할 수 있다. 이때 마찬가지로 세션의 특성을 SDP로 기술하여 개설자는 SIP프로토콜을 이용하여 참가자들에게 초청장을 보내게 된다. HTTP 버전 1.1 포맷으로 개설자와 참가자에 관한 정보가 포함된 SIP 메시지가 참가자에게 전달되면 그 참가자는 초청을 수락할 경우 연결이 설정되었다는 메시지를 개설자에게 보내고 수락하지 않을 경우 아무런 메시지도 보내지 않는다.[19] 참가자의 초청을 위해서 프락시(proxy) 서버를 운영할 수도 있는데 프락시 서버는 참가자의 정확한 위치를 알아내거나 SIP 메시지를 중계하는 역할을 수행한다.

### 3.2.6 실시간 스트리밍 프로토콜(RTSP)

RTSP[12]는 IETF MMUSIC(Multiparty Multimedia Session Control) 그룹에 의해 RFC2326으로 표준화 된 프로토콜로, 클라이언트/서버 환경에서 오디오나 비디오 데이터를 제공하는 서버로부터 저장된 정보를 실행시키고 이를 제어하는 기능을 제공하며, IP 네트워크 상에서 스트림 형태의 멀티미디어 데이터를 효과적으로 전송하기 위한 전송 제어 프로토콜이다. RTSP는 HTTP와 거의 유사한 프로토콜로서 표준의 대부분이 HTTP와 동일하다. RTSP 포맷의 형태는 HTTP 1.1과 유사하며 텍스트 기반의 프로토콜이다. HTTP는 TCP를 이용하여 시간적 제약이 없이 완전한 신뢰성을 요구하는 웹 페이지를 전송하는데 적합한 프로토콜이다. 그러므로 실시간성을 요구하는 멀티미디어 데이터의 전송 제어에는 RTSP가 권고되고 있다<sup>[14]</sup>.

RTSP를 위한 전송 프로토콜로는 UDP, IP 멀티캐스트, TCP 등이 이용될 수 있으나 앞서 설명한 것처럼 RTP는 실시간 데이터의 전송을 위한 프로토콜이므로 RTSP와 연동하여 사용할 수 있다. 이 때 RTP는 서버의 멀티미디어 파일을 클라이언트에게 단방향으로 전송하는 역할을 한다. 그러므로 서버의 멀티미디어 데이터를 클라이언트의 사이트에 나타내기 위해서는 스트림 엔진으로 RTSP와 RTP의 구현이 요구된다. RTP를 전송 프로토콜로 사용할 경우 RTP가 수정되거나 기능이 추가되어도 RTSP는 계속적으로 사용할 수 있는 장점이 있다.

RTSP는 멀티미디어 데이터를 저장하고 있는 서버와 데이터의 전송을 요청하는 클라이언트 구조로 구성되어 있다. 서버는 여러 클라이언트와 연결이 가능하며 동일한 미디어 파일을 여러 클라이언트에게 제공할 수 있어야 한다. 먼저 클라이언트가 데이터의 전송을 요청하면 서버는 클라이언트와의 연결을 설정한다. 연결 설정이 이루어지면 양쪽의 RTSP프로토콜은 RTSP 메시지를 이용하여 제어정보를 교환한다. 제어정보의 내용에는 미디어의 포맷, 관련된 요소들, 전송 방법, 전송포트 등을 포함한다. 이 때 이용되는 프로토콜은 신뢰성을 고려하여 TCP를 이용한다. 전송 제어정보를 교환한 후, 멀티미디어 데이터를 송수신하기 위해 양측에서는 RTP를 작동시키며 서버는 클라이언트로 지정한 파일을 패킷화하여 클라이언트로 전송한다.

## 4. QoS 제공을 위한 기술

본 절에서는 3절에서 설명한 프로토콜과 관련해서

구현에 관련된 기술들을 소개한다. 3절에서는 표준화된 프로토콜의 동작과 의미를 살펴 보았는데 본 절에서는 성능과 관련된, 즉 QoS를 향상시키기 위한 비표준화 된 기술들을 기술한다. 이들 기술은 크게 3가지 분야로 나누어 볼 수 있는데 흐름제어 기술, 에러 복구 기술, 그리고 마지막으로 다양한 네트워크에 산재된 수신자의 전송 속도에 알맞게 전송하는 기술이다.

#### 4.1 흐름 제어 기술

인터넷에서 자원의 예약이 없이 QoS를 보장하기란 어렵기 때문에 현재 인터넷의 조건을 가지고 좋은 품질을 제공하기 위한 최선의 방법은 종단간의 흐름제어이다. 즉, 사용 가능한 네트워크 대역폭을 측정하여 전송율을 동적으로 조절함으로써 네트워크 성능을 최대한 높이는 것이다. 이를 위한 많은 연구가 진행되고 있는데 대표적으로 INRIA의 IVS(INRIA Video-conferencing System)<sup>[60]</sup>가 있다.

IVS는 다음과 같은 방식으로 동작한다. 송신자는 주기적으로 측정된 수신자들의 패킷손실에 의해 평균 혼잡상태를 계산하고, 계산된 평균 혼잡상태에 따라서 전송율을 일정량만큼 선형적으로 증가 또는 지수적으로 감소시킨다. 이 때 증가량과 감소량은 네트워크 상태와는 관계없이 정적으로 정해진다는 단점이 있다. 즉 증가량이 너무 큰 경우, 갑작스런 데이터의 유입으로 자신을 포함한 다른 트래픽들에 많은 패킷 손실을 유발시키며, 반면에 증가량을 너무 작게 하면 네트워크 대역폭을 효과적으로 사용하지 못하는 문제가 발생하게 된다.

이러한 IVS 흐름제어 기법의 성능을 향상시키기 위해 본 고의 필자는 TCP의 동적인 전송 속도 조절 알고리즘을 응용한 TCP-like 흐름제어 기법<sup>[8]</sup>을 연구 개발하였다. TCP-like 알고리즘은 네트워크가 혼잡할 때는 IVS와 같이 지수적으로 전송 속도를 감소시키고, 네트워크가 무부하 상태로 감지되면 TCP와 같이 전송 속도를 두 개의 서로 다른 증가율로 증가시킨다. 즉, 전송 속도가 어느 정도 이하이면 TCP의 슬로우 스타트(slow start)와 같이 지수적으로 증가시키며, 어느 일정 속도 이상이면 TCP의 혼잡 회피 단계(congestion avoidance)처럼 선형적으로 증가시킨다. 또한 TCP-like 기법에서는 IVS와는 다르게 두 가지 네트워크 성능 변수인 패킷손실과 왕복시간(RTT)을 이용하여 네트워크의 상태를 파악한다.

#### 4.2 에러 복구 기술

실시간 멀티미디어 어플리케이션에 있어서 패킷손실과 패킷지연은 수신된 영상/음성 신호의 품질에 매

우 중요한 영향을 미치는 요소이다. 패킷손실은 주로 네트워크에서의 혼잡에 의해 주로 발생하기 때문에 품질을 급격히 저하시키게 되는데 특히 영상에 비하여 음성에 있어서의 패킷손실은 상대방의 음성을 알아들을 수 없을 정도로 나쁜 영향을 끼치게 된다. 더욱이 어느 정도 간격을 두고 발생하는 패킷손실은 품질에 덜 영향을 끼치는데 비해 연속적으로 발생하는 패킷손실은 상당한 품질의 저하를 가져온다. 품질에 영향을 주는 또 하나의 중요한 요소는 네트워크에서 발생하는 패킷지연이다. 품질을 높이려면 두 패킷 간의 도착시간의 차가 작아야 하며 제로(0)에 가까워야 한다.

대부분 품질 저하의 주요 원인은 네트워크에 있다. 이 문제점을 해결하기 위해 두 가지의 해결 방법이 있는데 그 중 한 가지는 실시간 트래픽을 지원하기 위하여 라우터에서 전송 프로토콜과 스케줄링 알고리즘을 변경하는 것이다. 그러나 좀더 보완해야 할 문제점들을 안고 있고 아직은 인터넷의 모든 라우터에 적용되지 못하고 있다. 두 번째는 응용 프로그램 단계에서 최선의 알고리즘을 구현함으로써 네트워크에 의해 발생하는 낮은 품질의 서비스를 극복하는 것이다. 이 방법에 의해 손실된 패킷을 복구할 수 있는 대표적인 제안으로는 ARQ(Automatic Repeat Request)와 FEC(Forward Error Correction)<sup>[61]</sup>이 있다. ARQ 기법은 수신자의 요청에 따라 재전송을 통하여 손실된 패킷을 복구하는 것인데 많은 통신 프로토콜에서 상용하고 있는 방법이다. 그러나 재전송에 의해 패킷이 수신자에게 전달될 경우 패킷지연이 발생하므로 실시간 데이터 전송에는 적합하지 못하다. 반면에 FEC 기법은 재전송에 의한 방법이 아니라, 송신자가 현재의 패킷에 에러를 복구할 수 있는 잉여 데이터를 덧붙여서 전송하면 수신자는 패킷손실이 발생했을 경우 잉여 데이터를 통해 복구하는 방법이다.

여러 가지 FEC 기법 중 한 가지 예<sup>[62]</sup>를 들면 다섯 개의 패킷마다 한 번의 잉여 데이터를 덧붙여서 전송하는 방법으로 다섯 개의 패킷마다 하나의 그룹 즉, 윈도우로 구분하고 윈도우의 첫번째 패킷에는 잉여 데이터를 덧붙이게 되다. 잉여 데이터를 생성하는 방법은 이전의 다섯 개의 패킷으로 exclusive-or 연산을 수행하는데 그 이유는 이전 패킷들과 서로 비교해서 다른 값만을 얻어내려는 것이다. 그림 7에서 만약 3번째와 9번째 패킷이 손실되었다고 가정하면 3번째 패킷을 복구하기 위해서는 6번째 패킷이 도착한 후에, 9번째 패킷을 복구하기 위해서는 11번째 패킷이 도착한 후에 복구할 수 있다. 그러나 한 윈도우 내에서 하나 이상 또는 연속적인 패킷손실이 발생했을 때는 복구할 수 없기 때문에 이를 해결하기 위해 많은 연구가

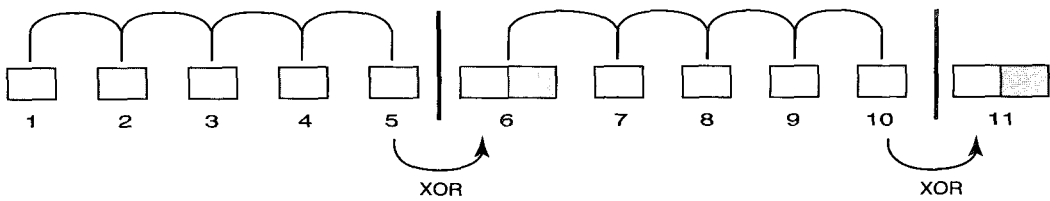


그림 7. 다섯 윈도우에서 잉여 데이터 생성

이루어지고 있다.

#### 4.3 다양한 네트워크 수신자를 위한 전송기술

최근에 멀티캐스트 연구와 연관되어 인터넷과 같은 이질적(heterogeneous) 네트워크 상에서 효율적인 그룹 통신을 제공하는 연구가 관심을 끌게 되었다. 인터넷의 이용자는 64Kbps 모뎀, 128Kbps ISDN과 10Mbps 이더넷 등 다양한 전송 네트워크 들에 연결되어 있을 수 있기 때문에 이러한 이질적 네트워크에 산재되어 있는 수신자들 모두를 만족시킬 수 있는 데이터의 전송 속도를 정하기 어렵다. 이와 같은 문제점을 해결하기 위한 방법으로 두 가지가 제안되고 있다.

첫째로, 계층적 코딩과 이를 위한 전송기술이다. 이미 H263과 MPEG-4의 표준안에 적용된 바 있는 계층적 압축기술은 종래의 기술과는 다르게 동영상 이미지를 기본 계층과 여러 개의 고품질 계층으로 구성된 데이터 스트림으로 나눠 생성한다. 이를 통해 수신자는 기본 계층에 더하여 자신의 수신 능력에 따라서 몇 개의 고품질 계층을 선택하여 동영상을 상영할 수 있다. 이 때 고품질 계층을 많이 수신할수록 고품질의 영상을 볼 수 있다. 계층적 코딩을 위한 전송기술은 수신자가 자신의 네트워크 체증상태에 따라 동적으로 수신할 계층의 개수를 결정하는 기법이다. 수신자의 상태가 체증상태면 수신하는 계층의 개수를 줄여 패킷 손실과 지연을 감소시키며, 수신 상태가 무 부하 상태가 되면 수신하는 계층의 개수를 증가시켜 고품질의 이미지를 상영할 수 있게 한다. 이러한 전송기술로는 이미 DSG[6], RLM[9]과 LVMR<sup>(4)</sup> 등의 기술이 제안되었고 실제 네트워크에서 구현하려는 연구가 진행중이다.

둘째로, 송신자와 수신자의 중간에서 트랜스레이터(translator)나 믹서(mixer)를 이용하여 수신자의 네트워크 속도에 맞추어 전송하는 방식이다. RTP는 트랜스레이터와 믹서의 기능을 지원하기 때문에 실시간 전송 프로토콜로 RTP를 사용하는 경우에만 가능하다. 트랜스레이터는 받은 패킷의 포맷을 단순히 다른 포맷으로 전환시켜 주는 역할을 한다. 가령 서로 다른 대역폭의 사용자들이 동시에 화상회의에 참여할 경우, 고품질의 높은 대역폭의 스트림을 저품질의 낮은 대역폭의 스트

림으로 바꾸어주는 역할을 한다. 이 때 멀티캐스트 주소 대신에 유니캐스트 주소를 사용하여 각각의 대역폭에 맞는 형식의 포맷으로 바꾸어 전송하면 사용자들마다 자신에게 알맞는 대역폭의 데이터를 받을 수 있다. 믹서는 트랜스레이터처럼 포맷을 바꾸는 것이 아니라 원래의 포맷을 유지하면서, 여러 개의 스트림을 하나의 스트림으로 합치는 역할을 한다. 이러한 방식은 합성이 가능한 오디오 데이터에 대해서는 매우 효율적인 방식이다.

## 5. 결 론

본 고에서는 실시간성의 멀티미디어 데이터를 전송하기 위한 스트리밍 기술과 연구동향에 관하여 간략히 소개하였다. 특히 본 고에서는 최근에 멀티미디어를 전송하기 위한 스트리밍 기술, 전송 프로토콜과 연구동향에 대해 알아봄으로써 멀티미디어에 관련된 현재의 기술 수준과 미비점 그리고 보완점 등을 소개하였다. 본문에서 언급했듯이 본 고의 저자는 현재의 문제점들을 보완하기 위해 새로운 연구와 이에 대한 표준화 작업 등이 꾸준히 계속되고 있으므로 인터넷의 발전과 더불어 멀티미디어 스트리밍에 관련된 압축, 전송, 재생기술과 같은 관련 분야도 더욱 발전해 갈 것으로 사료된다.

## 참 고 문 헌

- [1] C. Kenneth Miller, Addison Wesley, "Multicast Networking and Applications", September 1998.
- [2] ZDNet, <http://www.zdwebopedia.com/>
- [3] Jean-Chrysostome Bolot, Andres Vega-Garcia, "Control Mechanisms for Packet Audio in the Internet", INFOCOM, pp.232-239, March 1996.
- [4] Xue Li, Sanjoy Paul, Pramod Pancha and Mostafa Ammar, "Layered Video Multicast with Retransmission(LVMR) : Evaluation of Error Recovery, NOSSDAV, May 1997.
- [5] S. Blake, D.Black, M. Carlson, E.Davies, Z. Wang,



and W.Weiss, "An Architecture for Differentiated Services", FRC 2475, Dec. 1998.

[6] Shun Yan Cheung, Mostafa H. Ammar, and Xue Li, "On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution", INFOCOM, pp.553-560, March 1996.

[7] Daniel Ratton F. and Edmundo de Souza, "Efficient mechanisms for Recovering Voice Packets in the Internet", Globecom'99 pp1830-1837

[8] Seung-Gu Na and Jong-Suk Ahn, "TCP-like Flow Control Algorithm for Real-time Application", IEEE ICON2000, September 2000.

[9] Steven McCanne and Van Jacobson, "Receiver-driven Layered Multicast", SIGCOMM, pp.117-130, October 1996.

[10] Thierry Turletti, "The INRIA Videoconferencing System (IVS)", conneXions - The Interoperability Report, pp.20-24, October 1994.

[11] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC1889, January 1996.

[12] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC2326, April 1998.

[13] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP)", RFC2205, September 1997.

[14] 김두현, 김지용, 황승구, "차세대 웹상에서의 멀티미디어", 정보처리학회지, Vol.6, 1999.

[15] 송관호, 최선완 외 13인, "인터넷 전화/TV 표준연구", 연구보고서, NCA III-RER-97061, 1997.

[16] 김대영, "차세대 인터넷 기술동향", 정보과학회지, 제 17권 제 3호, 1999.

[17] T. Turletti and C. Huitema, "RTP Payload Format for H.261 Video Streams", RFC2032, October 1996.

[18] Kretschmer, Civanlar and Snyder, "RTP Payload Format for MPEG-2 AAC Streams", INTERNET DRAFT, June 1999.

[19] Dave Kosiur, Wiley Computer Publishing, "IP Multicasting: the complete guide to interactive corporate networks", 1998.

필자소개



나 승 구 (羅 承 九)  
 -1992            건국대학교 컴퓨터공학과(학사)  
 -1996            건국대학교 컴퓨터공학과(석사)  
 -1996~ 현재    동국대학교 컴퓨터공학과 박사과정  
 -주관심분야: 멀티캐스트 트래픽 흐름제어, 네트워크 보안



안 종 석 (安 鍾 奭)  
 -1983            서울대학교 전자공학과(학사)  
 -1983. ~ 1995    삼성전자 선임연구원  
 -1985.            한국과학기술원(KAIST) 전자공학과(학사)  
 -1995.            Univ. of Southern California 컴퓨터공학과(박사)  
 -1996. ~ 현재    동국대학교 조교수  
 -주관심분야: 컴퓨터 네트워크 시뮬레이션, 네트워크 프로토콜 개발