

## 위성 인터넷 액세스망 적용을 위한 실시간 전달 프로토콜의 구현 및 성능분석

이 계 상, \*진 광 자, \*안 재 영, \*오 덕 길  
동의대학교 정보통신공학과, \*한국전자통신연구원  
전화 : 051-890-1691 / 핸드폰 : 011-862-9712

### An Implementation of the Real Time Transport Protocol for Satellite Internet Access Networks

Kye-Sang Lee, \*Gwang-Ja Jin, \*Jae-Young Ahn, \*Deock-Gil Oh  
Donggeui University, \*ETRI  
E-mail : ksl@hyomin.donggeui.ac.kr

#### Abstract

Recently, with the advent of satellite communication technologies, the network access service through satellites has emerged as an alternative to existing ones. Of satellite access services including multimedia services, the high speed internet access service will be one of the important services. Particularly, the efficient transportation of real time internet traffic seems very essential.

The aim of this research is to examine the applicability of the IETF RTP protocol in the satellite access network. The RTP has been developed for the QoS-net transportation protocol of real time traffic over the existing best-effort internet. This paper summarizes the RTP protocol, and describes the architecture of the implementation of the RTP. And also, the compressed RTP header protocol is discussed for the applicability to the satellite link.

#### I. 서론

광대역 위성 멀티미디어 액세스망은 최신 위성 통신

기술을 이용한 다양한 망 구축과 이용 방안의 한 형태로 주목받고 있다. 특히, 광대역 위성 액세스망을 통한 초고속 인터넷 접속 서비스는 유선망을 통한 기존의 인터넷 접속 유력한 보완 수단으로, 격리 지역과 대도시 밀집 지역을 한번에 커버할 수 있는 장점을 갖는다. 특히, 위성 인터넷 접속 서비스의 주류를 형성할 것으로 전망되는 멀티미디어 서비스 제공을 위하여는, 기존 실시간 트래픽 전달 프로토콜의 위성 링크 구간에서의 적용성 검토 연구가 필요하다.

본 논문은 이를 위해 우선, 위성 인터넷 액세스망 구조를 개관한후, RTP 프로토콜 개요를 소개하고, 구현한 RTP 프로토콜 구조와 시험 결과를 요약한다. 다음, 위성 링크의 대역폭 효율을 높이기 위한 RTP 패킷 헤더 압축 프로토콜의 위성 링크 적용성을 논의한다.

#### II. 위성 액세스망과 실시간전달프로토콜

##### 2.1 위성 인터넷 액세스망

위성 통신 기술의 발전으로 위성채널은 그간의 망대 망 연결 수단에서, 이제는 망과 가입자를 연결하는 액세스망의 한 수단으로 부각되고 있다[1]. 여러 가지 서비스가 종합적으로 제공될 위성 멀티미디어 액세스망에서 특히 중요한 서비스는 초고속 인터넷 접속 서비

스가 될 것이다. 그림 1은 전형적인 위성 액세스망의 구성도를 나타낸다. 그림은 특히 상향과 하향 정보 전달이 모두 위성을 통해 이루어짐을 나타낸다. 이러한 위성 액세스망을 통해 실시간 인터넷 서비스를 제공할 때는 위성 링크의 고유한 특성, 즉, 긴 round-trip delay, 높은 링크 손실을 등이 실시간 전달 프로토콜에 미치는 영향이 고려되어야 할 것이다.

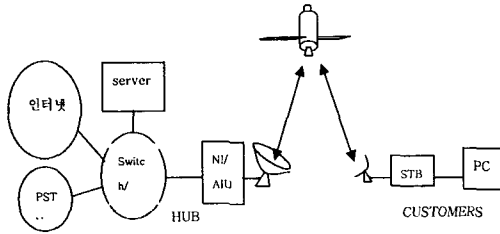


그림 1 위성 멀티미디어 액세스망 구조

## 2.2 실시간 전달 프로토콜 (RTP)

RTP 프로토콜은 실시간 트래픽 전달을 위한 새로운 프로토콜로서 인터넷 표준기구인 IETF의 avt 워킹 그룹에서 개발한 표준 프로토콜이다[2,3]. 이 프로토콜은 오디오/ 비디오 컨퍼런스를 염두에 두고 개발된 종단에 위치하는 응용 계층 프로토콜로서 실시간 트래픽의 엄격한 송수신 시간 제약성이 고려되었다.

RTP 프로토콜은 응용 계층 프레임 (Application Layer Framing) 구조를 갖는다. 이는 RTP가 실시간 응용 프로토콜에 공통으로 필요한 기본 동작과 표준 포맷 정도를 규정하는 간단한 프레임워크를 제공함을 의미한다. 따라서, 실제 하나의 완전한 실시간 응용 프로토콜은 RTP가 제공하는 프레임워크에 응용 고유의 기능들이 추가되어야 완성된다. RTP는 통상 UDP와 함께 실시간 멀티미디어 응용을 위한 트랜스포트 계층을 구성하며 UDP 프로토콜에 타이밍, 순서유지 기능을 추가하여 대부분 응용 프로토콜에 필요한 공통 기능을 제공한다.

RTP 프로토콜이 제공하는 기본 기능에는 타임 스탬프와 멀티캐스트의 지원을 들 수 있다. 타임 스탬프는 송신측에서 패킷들이 전송될 때 이 들간의 상대적 시간 간격이 수신측에서도 그대로 유지되어 재생될 수 있도록 하는 기능이다. 소스가 모든 메시지에 타임 스탬프를 표시하면 수신측에서는 이를 받아보고 패킷간 타이밍 정보를 추출하여 원만한 재생에 이용한다. 이 기능을 통해 패킷이 인터넷을 경과하면서 받는 지연 변이 (jitter)가 보정될 수 있다.

RTP 프로토콜은 오디오/ 비디오 컨퍼런스를 염두에

두고 개발된 프로토콜로 그 동작은 멀티캐스트 환경을 기본으로 하고 있다. 여러 사용자가 하나의 컨퍼런스에 참여하여 이룬 멀티캐스트 그룹에서 한 사용자의 발신 트래픽은 나머지 모든 사용자에게 전달된다. RTP 프로토콜과 함께 개발된 제어용 프로토콜인 RTCP (Real Time Control Protocol) 프로토콜도 멀티캐스트로 동작된다.

RTP는 모든 메시지에 동일한 단일 메시지 포맷을 갖는다. 실시간 트래픽의 전달에 효율적하도록 하고 대역폭의 낭비를 막고 지연 시간을 줄이는 등 오버헤드를 방지한 매우 간결한 헤더 포맷을 갖는다.

RTP 프로토콜은 실시간 트래픽을 전달하는 기능만을 담당하며, 프로토콜 제어에 필요한 별도의 기능은 RTCP 프로토콜을 통하여 달성한다. RTCP 프로토콜은 RTP 프로토콜과 함께 IETF avt 워킹 그룹에서 개발되었다[3]. RTP가 응용 데이터를 전달하는 단 하나의 메시지 유형만을 제공하므로 수신측으로 부터의 피드백 또는 응답을 위해 개발된 제어 프로토콜이다. RTP와 분리된 프로토콜이지만 밀접한 관계를 가지며 RTP와 짝을 이뤄 동작한다. 즉, RTP 프로토콜은 짝수의 포트 번호를 사용하고, RTCP 프로토콜은 통상 이보다 1이 더 큰 홀수의 포트 번호를 사용한다.

RTCP 프로토콜의 주 기능은 QoS 관리, 미디어간 동기, 소스 스트림 정보 식별 등이다. 이를 위해 5 개의 메시지 유형을 갖는다. 송신자 보고 메시지, 수신자 보고 메시지, 소스 식별 메시지, 종료 메시지, 응용 고유 메시지가 그 것이다. RTCP 프로토콜 메시지들 역시 RTP 패킷들과 함께 멀티캐스트된다.

RTP/ RTCP 프로토콜은 전달 품질이 보장되지 않는 현 best-effort 인터넷에서 실시간 트래픽의 시간 제약성을 어느 정도 충족시키고자 하는 프로토콜이다. 하지만, Integrated service와 Differentiated service와 같은 IP QoS 프로토콜에서와 같이 QoS 보장을 위한 어떤 자원 예약 기능은 수행되지 않음이 주의되어야 한다.

RTP/ RTCP 프로토콜은 최근 H.323의 일부 프로토콜로 쓰여 VoIP 서비스 제공에 응용되고 있고, 더욱 최근 들어서는 SIP/ SDP 프로토콜과 함께 쓰여 VoIP 뿐 아니라, 인터넷 방송과 원격 교육과 같은 스트리밍 서비스에도 활용되고 있고, 향후 넓은 범위의 A/V 컨퍼런스에 응용될 예정이다.

## III. 구현 구조 및 시험

### 3.1 구현 구조

프로토콜 구현 구조는 오디오나 비디오 등 각각의 미디어 스트림에 대해 별도의 세션을 구성하는 것을 가정하였다. 이들 서로 다른 미디어간 intermedia synchronization은 RTCP 메시지의 CNAME을 통해 달성한다. 하나의 세션은 다시 RTP 전송기와 수신기, 그리고 RTCP 전송기 및 수신기로 구성한다. 물론, 수신만 하는 경우 또는 송신만 하는 경우, 각각 RTP 수신기 또는 송신기만 동작될 수 있도록 한다. 그림 2는 이러한 실시간 트랜스포트 프로토콜 구현 구조를 도시한다.

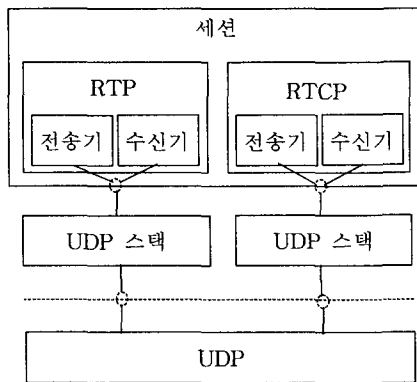


그림 2. 실시간 트랜스포트 프로토콜 구현 구조

세션 객체는 어떤 부가적인 기능을 담당하기 보다는 RTP 및 RTCP의 전송기 수신기 객체가 제공하는 기능을 encapsulation하여 API를 제공함으로써 상위 응용 프로그램 작성을 용이하게 함을 목적으로 한다. RTP 전송기는 RTP 패킷들을 생성하고, RTP 패킷을 전송하는 기능을 담당하며, RTP 수신기는 네트워크로부터 RTP 패킷을 수신하여 동기를 맞춰 상위 프로그램에 넘겨 주는 기능을 담당한다. RTP 전송기와 수신기는 각각 버퍼를 갖는다. 그림 3 참조.

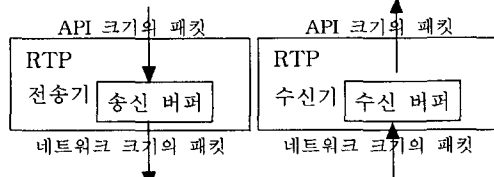


그림 3. RTP 전송기와 수신기 버퍼

RTP 전송기의 버퍼는 상위 응용 프로그램에서 페이로드를 포함한 데이터를 API 크기로 받아 일단 저장한 뒤 네트워크 전송에 맞는 단위로 변환하기 위한 장소로 쓰인다. RTP 수신기의 버퍼는 전송기 버퍼의 패킷 크기 변환 기능 이외에 인터넷 통신망에서 발생한 각 패킷의 지연변이를 흡수하는 기능을 담당한다.

RTP 전송기와 수신기는 동일한 소켓을 통해 하나의 UDP 프로토콜 스택을 공유한다. UDP 스택은 포트 번호, 소켓 번호 등을 모아 놓은 자료 구조이며, 소켓 프로시저를 간접 호출하여 RTP 및 RTCP 전송기와 수신기가 UDP 소켓을 쉽게 사용하도록 한다.

RTCP 프로토콜 스택도 RTP 스택과 유사하게 설계되었다. RTCP 전송기는 RTCP 메시지를 생성하여 전송하는 기능을 담당하며, 수신기는 RTCP 패킷을 수신하여 검사하고 유형별로 그 내용을 해독한다. RTCP 전송기와 수신기도 별도의 UDP 스택을 공유한다.

RTP 데이터 구조는 패킷 클래스, 패킷 헤더 클래스, 전송기 클래스 및 수신기 클래스로 구성된다.

RTP 패킷은 패킷 클래스와 헤더 클래스로 구성된다. 패킷 클래스는 실제 패킷 데이터를 갖는 메모리 영역을 가리키는 포인터와 패킷 헤더를 가리키는 포인터로 구성되며, 패킷의 크기 등 패킷에 관련된 일반 정보를 포함한다. 또한 이러한 패킷 정보에 접근할 수 있는 멤버함수를 제공한다. 패킷 헤더 클래스는 구조체로 정의되며 RTP 프로토콜 패킷 헤더의 여러 필드를 담고 있다.

RTP 전송기 클래스는 RTP 패킷을 만들고 UDP 스택을 이용하여 패킷을 전송하는 기능을 담당한다. 사용자 프로그램에서 RTP API를 통해 페이로드가 넘어 오면 이를 송신 버퍼에 잠시 보관하였다가 네트워크 패킷 크기에 맞추어 전송한다. 패킷 수신지에 관한 정보는 생성자 기능시 remoteHost와 remotePort 인수를 통해 얻는다. 전송기 클래스는 전송기 관련 여러 변수를 갖고 있으며, 또한, QoS 모니터링을 위해 RTCP 전송기와 수신지가 이 클래스의 변수에 접근할 수 있도록 한다. 패킷의 전송은 궁극적으로 UDP 스택을 통해 이루어지며 아울러 UDP 정보에 접근하는 방법도 제공한다.

RTP 수신기 클래스는 네트워크로부터 패킷을 수신하여 jitter 제거를 위한 입력 버퍼를 거쳐 패킷 재생 시각에 맞춰 패킷을 재생하는 과정에 필요한 기능들을 주요 멤버함수로 갖는다. 패킷을 수신하면 jitter 값을 갱신하고 사용자 프로그램에 패킷을 넘겨 줄 재생 시각이 되었으면 이를 시행한다. 어떤 소스로부터 최초의 패킷이 도착하면 이 소스를 수신기에 등록시키고 관련 정보를 초기화한다. QoS 모니터링을 위해 RTCP 프로토콜과 정보를 공유한다.

RTCP 데이터 구조도 RTP 데이터 구조와 유사하게 패킷 클래스, 패킷 헤더 클래스, 전송기 클래스 및 수신기 클래스로 구성된다.

### 3.2 구현 환경 및 시험

Red Hat 6.1 리눅스[4]가 탑재된 펜티엄 PC 상에서 C++ 언어와 GNU C 컴파일러 및 디버깅 툴[5]을 사용하였다. RTP/ RTCP 프로토콜 기능 구현 부분은 라이브러리화하고, 테스트 프로그램과 같은 응용 프로그램과 링크하여 사용하였다. 응용 프로그램 예제로서, PCM 코딩된 음성을 송수신하는 것을 가정하여, 20 ms 마다 160 바이트씩을 갖는 패킷을 주기적으로 송수신하는 테스트 프로그램을 작성하여, loop-back 인터페이스를 통한 동일 호스트 시험 및 LAN 시험을 완료하였다.

#### IV. 헤더 오버헤드의 분석

RTP 패킷의 전송에 필요한 총 헤더 오버헤드는 최소 40 바이트가 된다. 이것은 최소 12 바이트의 RTP 패킷 헤더와 8 바이트의 UDP 헤더, 그리고 20 바이트의 IP 헤더를 합한 것이다. 40 바이트의 헤더 오버헤드는 링크 대역폭의 이용률을 현저히 감소시킬 수 있다. 예로서, 앞 절에서 살핀 PCM 코딩 음성 전송의 경우, 20ms 마다 160 바이트의 음성 페이로드를 전달하기 위해 최소 총 40 바이트의 헤더 오버헤드는 20%의 bandwidth 이용률의 저하를 유발하게 될 것이다. 이러한 패킷 헤더 오버헤드는 하향에 비해 대역폭이 상대적으로 작은 상향 위성 링크의 이용도 성능의 저하에서 두드러질 것이다.

한편, 최근 저속도 링크를 통한 RTP 패킷의 전달시 헤더 오버헤드를 감소시키는 연구가 있었다[6]. 이 연구에서는 다이얼 옆 모뎀의 저속도 링크를 통한 RTP 패킷 전달시 헤더를 작은 크기로 압축하는 방식이 표준으로 제안되었다. 이 방식으로, 압축 헤더는 2 또는 4 바이트의 크기를 갖게 된다. 헤더의 압축은 기본적으로, 첫째, 어느 특정 세션 동안 모든 패킷의 헤더의 필드중 절반 정도가 값이 변하지 않는 점을 이용하고, 둘째, 값이 변하는 필드도 인접한 패킷간 그 차이 (2nd order difference)를 구하면 상수가 된다는 점을 이용하여 이루어진다. 즉, 헤더 compressor와 decompressor간에 session context 정보로 일차, 이차 difference 값을 공유하며, 실제 전송되는 값은 context 식별자 정도로 제한함으로써 헤더 압축을 달성한다.

이 방식에서는 패킷 손실 또는 오류 등으로 session context invalidation이 발생하면, decompressor가 compressor쪽으로 CONTEXT\_STATE 패킷을 피드백함으로써 이를 알린다. 이 점에서 이 방식은 왕복 지연 시간이 작은 링크에서 최적으로 동작하나, 위성 링크와 같이 왕복 지연 시간이 큰 링크에서는 그 효율성이

떨어진다고 할 수 있다.

따라서, 위성 링크에 압축 RTP 헤더 방식을 적용할 경우, context invalidation의 발생을 가능한 줄이고, 또 발생시 피드백에 크게 의존하지 않도록 헤더 압축 프로토콜을 개선할 필요가 있다. 최근 IETF에서 이에 대한 연구[7]가 진행되고 있으나, 일반적인 여러 시나리오에 대한 개별적인 제안으로, 이러한 제안 중 위성 링크에 적용을 위한 방식 선택에 대한 추가 연구가 요구된다.

#### V. 결론

본 논문에서는 RTP 프로토콜의 위성 액세스망에서의 적용성 검토를 위한 사전 연구로서 RTP 프로토콜을 소개하고, 이의 구현 구조를 요약하였다. 현재 프로토콜의 구현 기능에 대해 루프백 인터페이스를 통한 동일 시스템내에서 시험과 LAN을 통한 시험을 완료하였다.

또한, 상향 위성 링크의 대역폭 이용도를 높이기 위한 RTP 패킷 헤더 압축 방식의 적용성을 살펴보았다. 현재, 표준으로 제안된 헤더 압축 방식은 왕복 지연 시간이 큰 위성 링크에는 적절치 않은 것으로 연구되고 있으며, 향후 현재 진행중인 몇 가지 개선책 중 적절한 취사선택 연구 내지는 새로운 연구가 필요한 것으로 생각된다.

#### 참고문헌

- [1] Special Issue, "Satellite ATM Network Architecture", IEEE Comm. Mag., 1999.3
- [2] H. Schulzrinne, et. al., "RTP : A Transport Protocol for Real-Time Applications", IETF RFC 1889, 1996.1
- [3] H. Schulzrinne, et. al., "RTP Profile for Audio and Video Conference with Minimal Control", IETF RFC 1890, 1996.1
- [4] C. Negus, "Red Hat Linux Bible", IDG Books, 1999
- [5] R. Stones, et. al., "Beginning Linux Programming", Wrox, 1999
- [6] S. Casner, and et. al., "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", IETF RFC 2508, 1999.2
- [7] Tmima Koren and et. al., "Enhancements to IP/UDP/RTP Header Compression", draft-ietf-avt-crtcp-enhance-00.txt, 2000.7