

# 핸드폰에서 UDP-Lite Protocol 을 이용한 오디오 스트리밍 기법

류은석\*, 유혁

\*고려대학교 컴퓨터학과

e-mail : { esryu , hxy }@os.korea.ac.kr

## Audio streaming system on mobile phone using UDP-Lite protocol

Eun-Seok Ryu \*, Chuck Yoo

Dept. of Computer Science, Korea University

### 요 약

무선 데이터 통신이 하루가 다르게 증가하면서 무선망에서 핸드폰을 이용해 멀티미디어 데이터를 스트리밍하는 기술들이 연구되며 소개되어지고 있다. 이러한 기술을 연구하는데 있어서는 첫째로 데이터 전송 실험을 통한 무선 망에 대한 이해 및 특성 파악이 필요하고, 둘째로 이런 망 특성에 따른 적합한 스트리밍 기법 연구가 필요하다. 본 논문에서는 이러한 방법에 따라 망의 특성 파악에 관한 데이터를 기반으로 실험을 통해 EVRC 오디오 코덱을 이용한 핸드폰에서의 오디오 스트리밍에 있어 좀 더 나은 에러 대처 방법과 트랜스포트 프로토콜의 사용 기법을 제안하고 있다.

### 1. 서론

일반적으로 유선망에서의 오류 발생률은 통계적으로 보아 1%미만으로 알려져 있으나, 무선구간에서는 국내 전파 환경을 기준으로 5%~10%정도의 오류 발생을 보이는 것으로 나타났다.

유선망에 비해 상대적으로 높은 오류 발생률로 인해, 특히 실시간성을 가지는 스트리밍 시스템에서의 멀티미디어 데이터의 전송은 Error Resilient coding 기법들을 필요로 한다.

결국 시스템 설계에 있어 이러한 실시간성을 지켜 주기 위해서는 버퍼링(buffering) 시간을 단축하게 되고, 이로 인해 데이터 오류 발생시 재전송(retransmit)을 위한 충분한 시간을 확보하기 힘들게 되었다.

따라서, 서버 측에서 멀티미디어 데이터 전송 전에 미리 에러 복구를 위한 코드를 삽입하여 전송하고 이를 받는 클라이언트 측에서 에러 발생시에 이 코드를 이용하여 에러에 의한 데이터의 손실을 최소화 하는 여러 기법들이 연구되고 사용되었다.

하지만, 이에 대한 많은 연구들은 주로 CPU 성능이 충분한 PC 클라이언트를 대상으로 진행되어서 다소 복잡하고 redundancy 가 많은 단점을 가지고 있으며, codec specific 하지 않은 에러 내성 코딩(error resilient coding)기법들은 때로는 굳이 복구하지 않아도 될 데이터까지 복구를 위해 애쓰도록 한다.

이와 관련하여 transport layer 를 중심으로 연구되어진 방법 중 하나가 UDP-Lite 기법으로 UDP 입장에서

데이터 필드 전체에 대해 체크섬(checksum) 수행 후 잘못된 패킷(packet)을 버리는 방법 대신 상위 계층에서 헤더로 사용되는 부분에 대해서만 부분 체크섬을 함으로서 중요하지 않은 부분에 발생한 오류 데이터를 가진 패킷(packet)을 버리지 않고 상위 layer 로 옮기는 기법이 있다.

이는 사용하는 코덱이 Error resilient 한 특성을 가지고 있을 때, 데이터 필드에서의 1 bit error 등과 같이 작은 에러에 대해 UDP 계층(Layer)에서의 체크섬을 통한 버림(drop) 보다는 상위 계층에서 복구하여 사용할 경우 오디오의 음질과 비디오의 화질에 있어 좀 더 좋은 성능을 낼 수 있음을 보여주는 기법이다.

본 연구에서는 상대적으로 CPU power 가 작은 무선 이동 단말에서의 AOD(Audio On Demand) 및 멀티미디어 스트리밍 서비스를 위한 EVRC(Enhance Variable Rate Codec)를 이용한 audio streaming system 을 설계하고 구현하였으며, 이 시스템이 UDP-Lite protocol 위에서 동작될 경우 어느 정도의 시스템 성능 향상을 가져올 수 있는지 에뮬레이터 위에서 실험하고 그 결과를 제시한다.

### 2. 배경 설명 (Background)

#### 2.1 무선 네트워크에서의 UDP 프로토콜

멀티미디어 데이터 전송을 위해 주로 사용되는 UDP 프로토콜에서의 체크섬은 무선 네트워크에서 오

Packet Size	Version/Type	Timestamp	Rate series size	Rate series	Extension	EVRC Data
2 byte	2 byte	4 byte	2 byte		2 byte	

[표 1] Packet 구조

히려 데이터의 품질을 저하시킬 수 있다. 특히, 전송되는 데이터가 동영상일 경우에는 체크섬으로 인해 전송과정에서의 발생한 1bit 에러를 포함한 패킷까지 버리게 되므로 비디오의 경우에는 화면의 깨짐이 커지고, 오디오의 경우에는 무음이나 사운드의 왜곡이 일어난다. 즉, 기존의 UDP 프로토콜은 오늘날의 에러 내성 코덱(Error Resilient Codec)을 제대로 지원하고 있지 못하다.

2.2 UDP-Lite 의 개념

체크섬 에러에 의해 버려지는 패킷들을 최대한 살려 상위 계층으로 전달함으로써 화질이나 음질을 개선시킬 수 있다.

Source address	
Destination address	
zero	UDP Length
Source port	Destination Port
Length	Checksum

UDP Protocol

Source address	
Destination address	
zero	UDP Length
Source port	Destination port
Coverage	Checksum

UDP-Lite Protocol

[표 2] UDP Protocol 과 UDP-Lite Protocol

표 2 에서 보는 바와 같이 UDP-Lite 프로토콜은 Length 필드가 Coverage 필드로 바뀌어져 있다. Coverage 는 부분 체크섬을 위해 트랜스포트 계층에서 데이터에 해당하는 부분 중에 상위 계층에서 헤더로 사용되는 등의 중요한 부분의 길이를 말한다. 즉, 기존의 UDP 에서 데이터를 포함한 전체 길이였지만, UDP-Lite 에서는 부분 체크섬을 할 중요한 부분만의 길이를 넣어주면 된다.

3. 시스템 구조

3.1 System Overview

현재 대다수의 mobile phone 에서 사용되는 vocoder 인 EVRC 의 데이터는 각 프레임의 헤더 값인 rate 에 따라 데이터 길이가 가변적이다. 기존의 음성 통화를 위한 데이터 전송에는 라디오 링크(radio link)계층에서 RLP(Radio Link Protocol) non-transparent mode 일 경우 에러복구를 하게 된다. 하지만, transparent mode 일 경우 에러 발생에 대해 재전송이나 packet drop 을 하지 않고 상위 layer 로 데이터를 올려준다. 본 연구에서의 EVRC 를 이용한 사운드 스트리밍 시스템에서는 이렇게 상위 layer 로 올라온 데이터에 대해 EVRC 특성에 맞춘 Error recovery 를 시도한다.

오디오 코덱으로 EVRC 를 택한 것은 이미 대부분의 mobile phone 에서 하드웨어를 통해 EVRC 디코딩을 하기 때문에 그 하드웨어를 사용하기 위해서이다.

EVRC 는 rate 값이 쓰여있는 1byte 헤더 부분에 오

류가 발생할 경우 그 프레임 데이터 전체를 사용할 수 없으며 연속된 패킷 데이터에 대해서는 자칫하면 error propagation 을 초래한다. 따라서 CPU 연산이 많이 필요하고 redundancy 가 많은 FEC(Forward Error Correction)같은 방법을 이용하기 보다는 전체 데이터가 아닌 이러한 EVRC 프레임 rate 값을 따로 모은 rate series field 를 두고 이에 대해서만 error recovery 를 시도하는 방법을 사용한다.

3.2 Packet 구조

구현된 시스템에서는 EVRC 데이터를 효과적으로 스트리밍 하기 위해 표 1 에서 보는 바와 같은 Packet 구조로 전송한다.

Timestamp 필드는 오디오의 개별적 재생뿐 아니라 비디오와 함께 재생될 때, 동기화를 맞추기 위하여 4 byte 로 이루어져 있으며 ms 단위를 가진다. 현재 사용되고 있는 mobile phone 의 msm5100, msm5500 CPU 의 경우 그 연산처리 능력이 PC 에 비해 상대적으로 좋지 못하므로 Rate series 필드는 EVRC 데이터의 rate 값들을 각각 1byte 씩 표현한다.

EVRC rate 의 경우 1 에서 4 까지의 정수 값을 가지므로 여러 rate 값을 2bit 씩 할당하여 저장할 수 있으나 mobile phone 등에서 비디오 등과 함께 소프트웨어 (software)로 디코딩(decoding)될 때는 CPU power 에 매우 민감하기 때문에 byte 단위로 저장하였다.

본 논문에서는 Audio Streaming 만을 다루고 있으나, 실제 구현된 시스템에서는 비디오와 함께 동기화를 맞추며 mobile phone 에서 재생되고 있다. 따라서 bit 연산은 가급적 피하고 있다.

EVRC data 는 가변적일 수 있으나, 본 실험에서는 400ms 분량씩을 Packet 화 하였다. 따라서 모든 데이터가 8KHz EVRC Full rate 의 경우라고 가정하면 23byte 의 20ms 프레임(Frame)이 20 개 모인 것으로 460 byte data 가 된다. 따라서 전체 패킷 사이즈는 500byte 정도가 된다. 이는 실제 무선망(wireless network)에서 PPP 를 이용하기 때문에 단편화(fragmentation)를 고려한 사이즈이다.

패킷 구조가 위와 같이 이루어져 있으므로 UDP-Lite 프로토콜(Protocol)을 이용해서 전송하기 위해서는 Rate Series size 필드와 Extension 필드를 읽은 후 EVRC Data 를 제외한 헤더 부분의 길이 만큼만 체크섬 할 수 있도록 해야 한다.

또한, UDP-Lite 프로토콜을 이용해서 부분 체크섬을 가할 때는 header field 부분만을 검사하게 된다. 헤더 부분에 대해서도 에러복구 코드를 집어넣을 수 있겠으나 전체 데이터에 비해 얼마 되지 않는 헤더부분에서의 에러 발생 빈도가 상대적으로 낮기 때문에 가끔 일어날 수 있는 header field 에서의 에러 발생에 대해서는 재전송으로 처리하는 편이 더 나을 수 있다.

Data 필드에 속하는 EVRC 데이터는 raw data 그대

로를 사용하는데, 패킷 사이즈를 줄이기 위해 rate 값을 굳이 제거하지 않은 이유는 H/W 적인 EVRC 보코더(vocoder) 및 S/W 적인 디코더(decoder) 모두에서 데이터의 사용이 가능하도록 하기 위해서이다.

4. 성능 평가

휴대폰에서 재생되도록 구현된 Audio Streaming system 은 무선네트워크의 높은 에러 발생 비율 때문에 중간에 사운드가 끊기는 때가 많다. 에러 내성 특성을 이용한 코딩을 사용하였다고 하더라도 transport layer 및 그 이하 계층에서의 패킷에 대한 체크섬을 통해 버려진다면 상위에서는 사용할 수 없다. 따라서 앞에서 소개한 핸드폰의 취약한 CPU 성능과 EVRC의 특성에 맞춘 에러 대처 기법을 사용한 후 UDP-Lite 프로토콜 위에서 스트리밍 하는 기법을 통해, 어느 정도의 성능향상이 이루어지는지를 실험하였다.

유선망에 속한 서버에서 CDMA-2000 무선망으로 데이터를 전송할 때 발생하는 어려움은 실제 망에서 다양한 데이터들을 전송하며 망의 특성을 파악하여 통계를 내어서 이를 이용하였다[8].

4.1 채널 시뮬레이션(Channel Simulation)

먼저, 네트워크 상황을 시뮬레이션 하기 위해 그림 1 에서와 같이 서버 측과 클라이언트 측으로 나뉘어 모델링하고 구현하였다. 다음으로, UDP-Lite의 체크섬 기능 등을 포함한 UDP-Lite simulator 를 클라이언트 측에 포함시켰다.

서버 측에서는 PCM 데이터를 S/W EVRC Encoder 를 통해 인코딩 한 후, 표 1 에서 설명한 바에 따라 패킷화 한다.

클라이언트 측은 받아진 패킷으로부터 헤더를 읽어 동기화를 맞춘다. 그리고 디코딩 시에 UDP-Lite 를 통해 검증되어진 rate series 값들을 참조하여 데이터 필드의 EVRC 데이터를 재생한다.

무선 네트워크에서 발생한 에러가 포함된 데이터를 클라이언트 측까지 전달되기 위해서는 RLP(Radio Link Protocol)에서 Transparent mode 로 전송이 되어야 한다.

이러한 부분을 시뮬레이션 하기 위해 서버 측에서 만들어진 데이터에 에러를 랜덤(Random)하게 발생시킨 후 이러한 데이터를 모아서 저장하였다.

이 때, 에러를 발생 시키는 비율은 위에서 언급한 대로 실제 망에서의 다양한 실험을 통해 측정되었다.

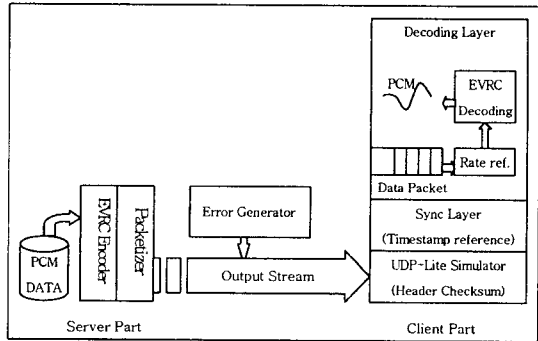
Packet Size (Byte)	Send Interval (ms)	Recv Interval (ms)	Correct (%)	Send Interval (ms)	Recv Interval (ms)	Correct (%)
300	60	66	92.6	30	35	93.2
350	70	73	95.2	35	40	97.2
400	80	88	86.2	40	50	85.8
450	90	99	92.8	45	59	88.6
500	100	162	82.4	50	55	94.0

550	110	119	93.0	55	65	94.6
600	120	129	92.8	60	70	88.4
650	130	139	86.0	65	83	83.8
700	140	154	89.6	70	79	90.4
750	150	165	91.2	75	95	82.8
1,000	200	209	94.6	100	112	89.0
1,500	300	317	95.4	150	167	90.2
전송속도	40,000 bps			80,000 bps		

[표 3] 유선⇒무선망으로의 데이터 전송 측정 결과

표 3 은 CDMA-2000 망에서 측정된 실험 결과이다 [11]. 본 논문에서 제안된 방식은 패킷 사이즈가 약 450Byte 에서 550Byte 정도가 되며 약 400ms 마다 전송을 하는 방식이다. 망 실험에 있어서 데이터를 전송하는 시간 간격에 따라서도 정확도(Correctness)가 조금씩은 차이가 있는 것으로 나타났으나 대략적으로 90% 정도였고, 본 실험에서는 이 정확도를 기반으로 에러율을 조정하며 그림 1 의 Error Generator 를 구현했다. 조금 더 많은 실험과 데이터를 모은다면 시뮬레이션의 정확도가 더 오를 수 있을 것으로 본다.

이렇게 저장된 데이터를 가지고 한번은 제안된 시스템에서 실험을 하고, 다른 한번은 에러 대처 기법이 없고 UDP-Lite 프로토콜을 사용하지 않은 기존의 방식에 의한 실험을 하였다.



[그림 1] Simulator System Architecture

5. 결과 및 고찰

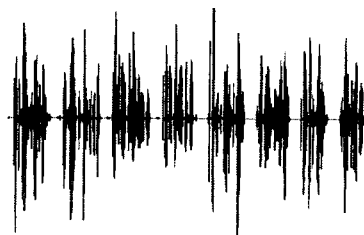


그림 2-A. Original PCM Data

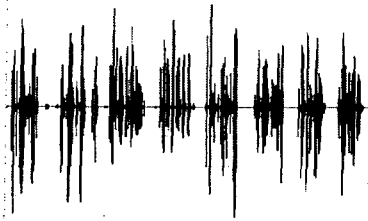


그림 2-B. Streaming over UDP

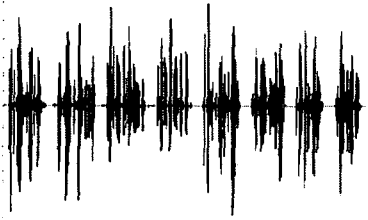


그림 2-C. Streaming over UDP-Lite

위의 그림 2-B 그래프에서 보는 바와 같이 에러에 대처하지 못하면서 udp-lite 프로토콜을 이용하지 않고 UDP 프로토콜을 이용한 경우의 시뮬레이션에서는 400ms 단위의 무음 구간이 중간 중간에 있다. 즉, 좋은 음질을 보일 수 없다. 그에 비해 그림 2-C 그래프는 본 연구에서 제안된 에러 대처 방법과 트랜스포트 계층의 프로토콜을 사용한 것으로 UDP-Lite의 coverage 필드에 해당하는 중요한 헤더 부분에 이상이 생기지 않는 한 패킷 버림이 없으므로 그림 2-B에 비해 원본 데이터인 그림 2-A에 더 가까운 PCM(Pulse Code Modulation) 파형을 가지고 있다. EVRC 데이터의 경우는 사실 약간의 데이터 값의 차이가 디코딩 되었을 때 상대적으로 많은 오류를 일으키는 것이 사실이다. 하지만, 그렇다고 하더라도 그 패킷을 그냥 버리는 것 보다는 좋은 성능을 보이고 있다. 좀 더 좋은 기능을 위해 에러가 있는 패킷을 복원하였을 때는 노이즈(Noise) 제거 필터 등을 통과하게 하는 등의 기법도 사용될 수 있다.

물론, 시스템을 제작하는 방법에 따라 한번에 패킷화하는 EVRC 데이터 사이즈를 줄이거나 충분한 버퍼링을 통해 재전송하는 방법으로 패킷 에러에 대처하는 방법도 있을 수 있다. 하지만, mobile phone에서는 메모리가 충분히 지원되기 힘들기 때문에 오디오, 비디오가 함께 스트리밍 될 때 충분한 버퍼링을 할 수 없다. 게다가 무선 네트워크의 특성 중 하나인 End to End Delay가 유선 네트워크에 비해 크다는 특성 및 데이터의 수신이 기지국에서의 버퍼링으로 인해 비교적 일정한 간격으로 이루어지지 않는 점 때문에 패킷 재전송은 더욱 어렵다.

또한 패킷 사이즈를 줄이거나 인터리빙(Interleaving) 방식 등을 사용하여 무음 구간을 좀 더 줄여볼 수는 있으나, 단순히 패킷 사이즈만을 줄이는 방법은 더 많은 전송으로 인해 데이터 사이즈에 비해 헤더 사이즈의 증가를 가져와서 결국 throughput의 감소를 가져온다.

FEC 등의 방법은 논문의 앞 부분에서 언급했듯이 패킷 사이즈의 증가와 너무 많은 CPU 사용으로 인해 현재의 mobile phone에서는 사용이 힘들다. 다만, 무선망의 특성인 burst error를 막기 위해 인터리빙 기법을 사용하는 것은 성능 향상을 가져 올 수 있다.

## 6. 결론 및 향후 과제

이상의 실험을 통해 EVRC를 이용하여 mobile phone에서 오디오 스트리밍을 할 때, 좀 더 나은 에러 대처 방법과 트랜스포트 프로토콜의 사용이 있음을 알 수 있었다.

앞으로는 구현은 되어 있으나 아직 데이터를 측정하지 못하여 본 논문에서 제시하지 못한 A/V 데이터를 스트리밍 했을 때의 실험 결과를 측정하고자 한다.

그리고 EVRC 특성상 SNR보다는 실험 결과에 대해 PESQ나 MOS 등에 의한 좀 더 객관적인 분석을 하려 한다. 또한, 좀 더 자세한 무선 망에 대한 특성 파악 실험을 통해 상대적으로 에러율이 높은 무선 망에서 좀 더 나은 멀티미디어 스트리밍 기법을 제시하고자 한다.

## 참고문헌

- [1] Ken C. Pholmann, "Principles of Digital Audio", McGraw-Hill
- [2] TIA/EIA/IS-127, "Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems", January 1997.
- [3] Lars-Åke Larzon, Mikael Degermark, and Stephen Pink, "Efficient use of Wireless Bandwidth for Multimedia Applications", In proceedings of the Sixth IEEE International Workshop on Mobile Multimedia Communications (MOMUC), 1999.
- [4] Lars-Åke Larzon, Mikael Degermark, and Stephen Pink, "UDP Lite for Real-Time Multimedia Applications", In proceedings of the IEEE International Conference of Communications (ICC), 1999.
- [5] Lars-Åke Larzon, Hans Hannu, Lars-Erik Jonsson, and Krister Svanbro, "Efficient Transport of Voice over IP over Cellular links", In proceedings of the IEEE Global Telecommunications Conference, 2000.
- [6] Amoolya Singh, Almudena Konrad, Anthony D, Joseph, "Performance Evaluation of UDP Lite for Cellular Video", NOSSDAV, 2001.
- [7] Lars-Åke Larzon, and Stephen Pink, "Requirements on the TCP/IP Protocol Stack for Real-Time Communication in Wireless Environments", In proceedings of international Workshop on QoS in Multiservice IP Networks, 2001.
- [8] 김일진, 정진환, 유혁, "효율적인 멀티미디어 데이터 전송을 위한 CDMA-2000 망의 특성 분석 연구", 정보과학회 학술대회, 2001.