화상 전화 단말기에서의 효율적인 오디오/비디오 동기화 방법

김찬우, 박성준, 서광덕
{chanwcom, sjpark0420}@lge.com, kdseo@dragon.yonsei.ac.kr

KISS Korea Computer Congress 2005

[1]Chanwoo Kim, [1]Seong-Jun Park, and [2]Kwang-deok Seo
[1]LG Electronics Handset Lab., 60-39, Kasan-dong, Kumchon-gu, Seoul, 153-801, Korea
[2]Computer & Telecom. Engineering Div., Yonsei Univ., Heugup-myon, Wonju, Gangwondo, 220-710, Korea

요 약

본 논문에서는 멀티미디어 데이터 전송시, 오디오와 비디오 데이터를 효율적으로 동기화·시키는 방법에 대해서 설명한다. 본 연구에서는 간단한 구조와 적은 연산량으로 동기화 알고리즘을 구현하여 내장형 시스템을 위한 프로세서에 적합하도록 고정 소수점으로 구현하는 방법을 얻게 되었다. OMAP 1510과 MSM5500으로 이루어진 화상 전화 단말기를 이용하여 실험한 결과 만족스러운 결과를 얻을 수 있었다.

1. 서론

근래에 들어서 멀티미디어 서비스에 대한 소비자들의 요구의 급격한 확산과 빠른 멀티미디어 프로세서들의 발전에 힘입어, 많은 멀티미디어 서비스들이 등장하게 되었다. 화상 전화 (비디오 Telephony), 주문형 비디오 시스템 (비디오-on-Demand)와 같은 것을 예로 들 수 있다. 이러한 멀티미디어 시스템에서는 오디오 신호와 영상 신호가 각각 Real-Time Transport Protocol (RTP)에 실려서 전송이 되게 된다. 본 논문에서는 RTP에 실려서 전달되는 오디오/비디오 신호간의 동기화를 기존 방법보다 매우 간단하게 처리하는 방법에 대해 제안한다. 본 논문의 구성은 아래와 같다. 2절에서 RTP를 이용해 오디오/비디오 데이터를 전송할 경우 기존의 동기화 방법에 대해서 설명한다. 그리고 3절에서 제안된 방법에 대해서 설명할 것이다. 4절에서는 구현된 전체 화상 전화 시스템에 대해서 설명할 것이며 5절에서 요약된 결론을 제시한다.

2. 기존의 멀티미디어 데이터 동기화 방법

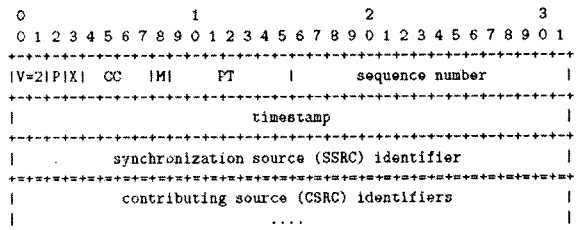

그림 1. RTP 패킷의 구조 [1]

멀티미디어 전송을 위해서 일반적으로 사용되는 프로토콜이 Real-Time Transport Protocol (RTP)과 RTP Control Protocol (RTCP)이다 [1]. 그림 1 은 RTP 프로토콜의 형태를 보여준다. RTP는 멀티미디어 데이터의 실시간 전송을 위하여 사용되는 프로토콜로서, sequence number나 RTP timestamp등 실시간 전송에 필요한 부가적인 정보를 헤더에 포함하고 있다.

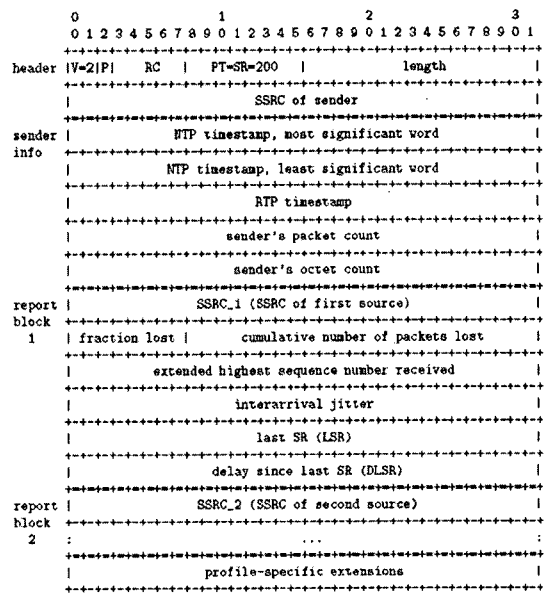

그림 2. RTCP SR 패킷의 구조

RTCP는 RTP를 제어하기 위해서 사용되는 프로토콜로서 Session Description (SDES), Sender Report (SR), Receiver Report (RR), Application (APP), 그리고 BYE의 다섯 가지 패킷 종류가 있다 [1]. RTP 패킷에 포함되어 있는 RTP timestamp는 유료 부하에 실린 데이터의 샘플링된 시간에 관한 정보를 담고

있다. 같은 RTP 세션 내에서는 RTP timestamp를 가지고 패킷의 전후 시간 관계를 알 수 있으나, 오디오와 비디오가 동시에 전송되는 화상 전화와 같은 경우와 같이 서로 다른 RTP 세션이 전송될 경우 각각의 RTP session에 실려 있는 RTP timestamp만을 가지고서는 동기화를 맞출 수 없다. 이는 각 세션의 RTP timestamp 값은 난수 (Random Number)로부터 시작이 되며, 또한 매체 마다 샘플링 레이트가 다르기 때문에 일정 시간마다 RTP Timestamp가 증가하는 비율이 매체 마다 다르기 때문이다 [1].

따라서 이와 같은 경우 오디오와 비디오의 동기화를 위해서 RTCP중 SR 패킷을 이용하게 된다 [1]. 그림 2. 는 RTCP SR 패킷의 구조를 보여 준다. 여기에서 보여진 바와 같이 RTCP SR 패킷은 RTP timestamp이외에도 절대적인 시간의 기준으로 활용될 수 있는, 즉 동기화의 기준으로 사용될 수 있는, NTP timestamp도 같이 싣고 있다 [2].

RTCP SR 패킷의 RTP timestamp와 NTP timestamp와의 관계를 조사하면 RTP 패킷으로부터도 그 패킷에 실려져 있는 RTP timestamp정보를 이용하여, 그 패킷의 유료부하가 샘플링 된 절대적인 시간을 알아낼 수 있으며, 이를 이용해 서로 다른 RTP 세션으로 전달되는 매체를 동기화 시킬 수 있게 된다.

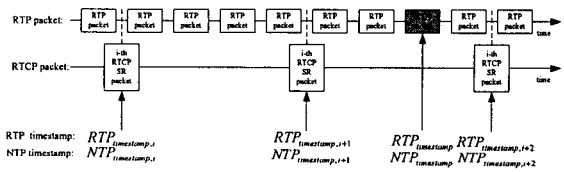

그림 3. RTP 패킷, RTCP 패킷, RTP timestamp, 그리고 NTP timestamp

그림 3. 은 RTCP 패킷이 i번째, $i+1$ 번째, 그리고 $i+2$ 번째가 차례로 들어오고 RTP 패킷에 데이터가 실려오는 일반적인 경우를 보여주고 있다. 이 그림에서, 음영으로 표시된 RTP 패킷의 경우, 절대시간을 기준으로 한 timestamp인 $NRTP_{timestamp}$는 그 패킷에 실려 있는 RTP timestamp인 $RTP_{timestamp}$ 와 다음의 (1)을 통해서 얻을 수 있다.

$$NRTP_{timestamp} = NTP_{timestamp,i+1} + \frac{\Delta NTP_{timestamp}}{\Delta RTP_{timestamp}} \bullet (RTP_{timestamp} - RTP_{timestamp,i+1}) \quad (1)$$

여기서 ,

$$\Delta NTP_{timestamp} = NTP_{timestamp,i+1} - NTP_{timestamp,i} \quad (2)$$

$$\Delta RTP_{timestamp} = RTP_{timestamp,i+1} - RTP_{timestamp,i} \quad (3)$$

이고 $RTP_{timestamp,i}$ 와 $RTP_{timestamp,i+1}$ 은 각각 i 번째와 $i+1$ 번째 RTCP 패킷에 실려 있는 RTP timestamp를 나타낸다. 또한 마찬가지로, $NTP_{timestamp,i}$ 와 $NTP_{timestamp,i+1}$ 는 i 번째와 $i+1$ 번째 RTCP 패킷에 실려 있는 NTP timestamp를 나타낸다 위의 $\Delta NTP_{timestamp}$ 와 $\Delta RTP_{timestamp}$ 로 나온 것은 두 RTCP SR 패킷에서의 NTP timestamp와 RTP timestamp의 차로부터 얻은 값이다. 위의 값으로 $NTP_{timestamp}$ 값을 구하면 그 값은 매체의 샘플링 레이트에 의존하지 않는 값을 얻을 수 있다.

3. 제안된 방법

일반적인 경우, 화상 전화에서 비디오과 오디오 데이터는 통화가 성립되어서 종결되기 전까지 각각 정해진 샘플링 레이트를 유지하는 경우가 대부분이다. 본 논문에서 제안된 방법은 이 경우와 같이 오디오와 비디오의 RTP 세션의 데이터가 고정된 샘플링 레이트를 가질 경우에 계산량을 줄이고 구현 구조를 간단하게 하는 것에 관한 것이다.

보다 구체적인 설명을 위해서, 일례로 비디오 데이터의 샘플링 레이트는 90 *kHz* 오디오의 샘플링 레이트는 8 *kHz*라고 가정하자.

이 경우 (1)에서 비디오의 경우는

$$\frac{\Delta NTP_{time\_stamp}}{\Delta RTP_{time\_stamp}} = \frac{2^{32}}{90k} \approx 47721.8588444 \quad (4)$$

가 되며 오디오의 경우 샘플링 레이트가 8 *kHz*일 경우

$$\frac{\Delta NTP_{time\_stamp}}{\Delta RTP_{time\_stamp}} = \frac{2^{32}}{8k} \approx 536870.912 \quad (5)$$

가 된다.

여기서 나눗셈을 한 것인 NTP timestamp의 소수 부분인 2^{-32} 만큼 증가분을 기반으로 해서 1초에 NTP timestamp는 2^{32} 만큼 증가를 하는 동안에 RTP timestamp값은 매체의 샘플링 레이트만큼 증가를 하는 데에 있다. 이와 같이 되는 이유는 NTP timestamp는 64-bit 변수로서 32-bit의 정수부분과 32-bit의 소수 부분으로 나누어져 있기 때문이다 [2].

(1)에서 비율을 매번 RTCP SR 패킷을 받을 때 마다 나눗셈을 해 주지 않고 이 정해진 값으로 사용해도 된다. 따라서 식 (1)에서는 RTCP에서 64 bit로 되어 있는 NTP timestamp 값을 매번 받아서 그것으로 나눗셈을 하지 않고 간단히 미리 정해진 값을 이용을 함으로써 연산량을 크게 줄이고 구현의 편리함도 도모할 수 있다. 위의 설명이 여기에서의 논문에 대한 설명이며 이를 이용해서 (1) 식에 대입해서 서로 비교를 할 수 있는 timestamp값을 가지고서 오디오가 빠르게 패킷이 도착이 되었는지, 비디오가 빠르게 패킷에 도착이 되어 있는지를 확인해서 버퍼에 들어가 있는 데이터들을 동기화가 되도록 오디오비디오의 동기화를 맞출 수 있다.

계산량은 고정 소수점 연산을 활용함으로써 보다 향상시킬 수 있다 [3]. 위의 식을 구현하고자 하는 프로세서가 부동 소수점 유닛 (Floating Point Unit)이 없는 고정 소수점 유닛 (Fixed Point Unit) ALU만 가지고 있는 일반적인 내장형 시스템용 마이크로 프로세서 [4]와 같은 경우, 위의 분수부 전체를 소수점 이하를 나타내는 32-bit 수로 생각할 수 있다. 이 것은 분자의 단위를 NTP timestamp도 1초를 기반으로 했을 경우를 가정한 것이다.

이 경우 고정 소수점 연산만으로도 계산을 할 수 있다. 특정한 예로 삼은 비디오의 샘플링 레이트는 90 kHz 이고 오디오의 샘플링 레이트는 8 kHz인 경우를 살펴 보자. 여기에서 적당히 스케일링을 하면 비디오와 오디오를 위한 (1)의 식이 각각 아래의 (6)와 (7)로 표현된다. 아래의 예는 오버플로우에 대한 고려와 SQNR을 작도록 하도록 고정

소수점 스케일링 (Fixed-point Scaling)을 하여 (1)을 표현한 것이다. 이하 아래에 나오는 표기법에서 Q.X 라고 표시되어 있는 것은 해당 변수에서 소수 부분이 X bit만큼 할당 되어 있다는 것을 의미한다.

$$NRTP_{timestamp} = (NTP_{timestamp,i+2})/*Q.10*/ + (47722/*Q.32*/ *((RTP_{timestamp,i+1} - RTP_{timestamp,i}) >> 10 /*Q.-10*/)) >> 12 \quad (6)$$

여기서 $NTP_{timestamp}$ 에 관계된 값은 (6)에 보인 바와 같이 Q.10 의 정밀도를 가지도록 하여 2^{-10} 초의 정밀도를 가지도록 하면 충분하다. $NTP_{timestamp}$ 는 Q.10으로 만들어야 하는데, NTP timestamp가 64-bit 값이므로 NTP timestamp의 정수 부분 (integer part) 중 하위 22-bit를 상위 22-bit에, 그리고 NTP timestamp의 FRAC part중 상위 10-bit를 하위 10-bit에 순차적으로 넣어주는 연산을 하면 Q.10에 해당하는 변수로 만들 수 있다.

위의 식으로 계산하면 나온 integer값 중 상위 22 bit 는 integer part를 나타내며 하위 10 bit 는 fractional part를 나타냄을 알 수 있다.

마찬가지로 오디오의 경우는 다음과 같이 해 줄 수 있다.

$$NRTP_{time\_stamp} = (NTP_{time\_stamp,i+2})/*Q.10*/ + (16777/*Q.27*/ *((RTP_{time\_stamp,i+1} - RTP_{time\_stamp,i}) >> 4/*Q.-4*/) >> 13 \quad (7)$$

이상의 (6), (7) 에서 알 수 있듯이, 동기화를 위한 수식이 나눗셈연산이나 64-bit 곱셈 연산 없이 32-bit 곱셈연산과 덧셈 연산, 그리고 쉬프트 연산만으로 효율적으로 이루어졌음을 알 수 있다.

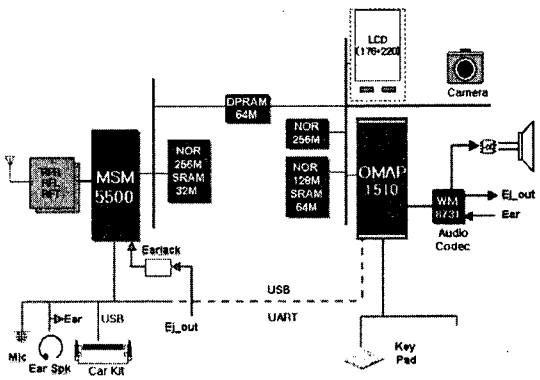

그림 4. 하드웨어 구조도

4. 시스템 구성

그림 4는 개발 된 화상 전화 시스템의 전체적인 구성도를 나타낸다. 베이스 밴드 모뎀은 Qualcomm사의 CDMA 1x EV-DO 용인 MSM5500이 사용되었으며, 애플리케이션 프로세서는 TI사의 OMAP1510이 사용되었다. 비디오 코덱의 경우는 OMAP1510내에 포함되어 있는 C5510 DSP 에서 인코딩/디코딩이 수행되며, 오디오 코덱의 경우는 베이스 밴드 모뎀으로 Inter-Processor Communication (IPC)를 통해서 전송된 후 디코딩 되어진다. 본 화상 전화 단말의 경우 비디오 코덱은 H.263 알고리즘이 사용되었으며, 오디오 코덱의 경우 QCELP 알고리즘이 사용되었다.

그림 5. 는 소프트웨어 구조도를 보여주고 있다. OMAP 1510 프로세서에서 비디오 코덱을 제외한 프로토콜 처리는 ARM925T 프로세서에서 처리된다. 따라서 RTP, RTCP 프로토콜과 본 논문에서 제안된 동기화 알고리즘도 ARM 프로세서에서 처리되도록 되어있다. ARM 프로세서는 Nucleus Plus가 OS로서 사용되었다.

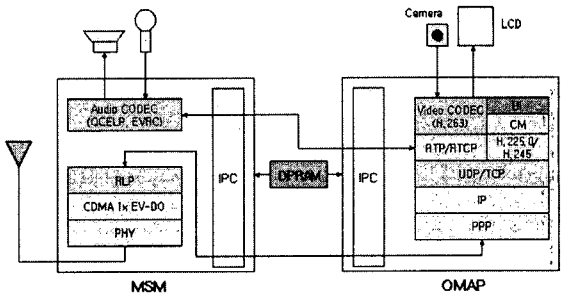

그림 5. 소프트웨어 구조도

실제 동작의 검증을 위하여 5번의 10 분간의 통화를 하여서 동기화 정도를 측정하였다. 디코더에 패킷을 전달하는 시간을 기준으로 하였을 때 오디오와 비디오 패킷의 동기화에서 0.5 초 이상의 오차가 발생하는 경우는 전체 패킷의 2 % 미만이었다. 휴대 전화에서의 오디오/비디오 동기화를 위한 본 알고리즘은 현재 특허로 출원중이다 [5].

5. 결론

본 논문에서는 RTP, RTCP 패킷을 이용하여 동기화를 구현하는 화상 전화 단말기에서 매체의 샘플링 레이트가 RTP 세션 동안 일정할 경우 매우 적은 연산량으로 동기화를 구현 방법을 제시함으로써 실제적인 응용에서 매우 효과적으로 활용될 수 있는 방법을 제시하였다. 제안된 방법의 핵심은 송신 쪽에서 일정한 샘플링 레이트로 오디오와 비디오가 샘플링 되었을 경우 성능상의 저하 없이 연산량을 획기적으로 적게 하면서 동기화를 맞출 수 있다는 것이다.

6. 참조 문헌

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Real-time transport protocol," *RFC 3550*, IETF, July 2003.

[2] D. Mills, "Network time protocol (version 3) specification, implementation," *RFC 1305*, IETF, March. 1992.

[3] S. Kim and W. Sung, "A floating-point to fixed-point assembly program translator for the TMS 320C25," *IEEE Trans. Circuits and Sys.*, Vol. 41, No. 11, pp. 730-739, Nov. 1994.

[4] S. Furber, *ARM System Architecture,* Harlow, UK: Addsion-Wesley, 1996.

[5] 김찬우, 서광덕, "휴대단말기의 비디오 오디오 동기 장치 및 방법", 대한민국, 출원번호 2004-0052619, 출원일자 2004.7.7.