

## 라이브 비디오 스트리밍을 지원하는 SIP 기반 단말기의 소프트웨어 시스템 분석

김혜선<sup>○</sup>, 황기태\*, 이재운\*, 김남윤\*\*

한성대학교 컴퓨터공학부\*

한성대학교 정보공학부\*\*

{happyi00<sup>○</sup>, calafk\*, jmlee\*, nykim\*\*}@hansung.ac.kr

### Analysis of software system for the SIP-based device supporting live video streaming

Hyesun Kim<sup>○</sup>, Kitae Hwang\*, Jaemoon Lee\*, Namyun Kim\*\*

Division of Computer Science and Engineering, Hansung University\*

Division of Information Engineering, Hansung University\*\*

#### 요 약

본 논문에서는 라이브 비디오 스트리밍을 지원하는 SIP 기반 단말기의 소프트웨어 구조와 설계에 대해 기술하였다. SIP 기반 단말기의 소프트웨어 시스템 구조를 분석하여 SIP 세션 설정 과정과 비디오 스트리밍 알고리즘을 설명하였으며 향후 연구로 임베디드 환경으로 포팅하기 위해 설계의 정확성을 검증하고 성능 평가를 위한 사전 실험을 수행하였다.

#### 1. 서 론

SIP(Session Initiation Protocol)[1]은 차세대 유/무선 네트워크 상에서 사용자 단말들 간에 멀티미디어 통신 세션을 설정하기 위한 시그널링 프로토콜로 IETF(Internet Engineering Task Force)에서 표준으로 채택되었으며, 다양한 부가 통신 서비스들을 구현하는 데 없어서는 안 될 핵심 프로토콜로 간주되고 있다. 이미 VoIP(Voice Over IP) 기반의 음성통신 서비스 시장이 형성되는 등 SIP 기반의 멀티미디어 통신 시장은 더욱 더 확대될 것으로 예상된다.[2]

한편 보안, 엔터테인먼트 등의 목적으로 IP 기반 카메라의 사용이 늘어나고 있다. 현재 개발되는 IP 기반 카메라는 HTTP를 기반으로 하므로 이동 중인 상황에서 세션을 유지하기 힘든 어려움이 있다. 그러나 SIP는 이동 중에도 단말 간 세션을 지속적으로 유지할 수 있으므로 SIP로의 전환이 불가피하다.

본 논문에서는 화상 카메라를 장착하고 비디오 스트리밍을 지원하는 SIP 기반 단말기의 소프트웨어 구조와 설계에 대해 기술하고자 한다.

#### 2. SIP 시스템

SIP는 크게 SIP UA(User Agent)와 SIP 서버들로 구성되며 전형적인 구성은 그림 1과 같다.[3]

SIP UA란 사용자가 콜(call)을 연결할 수 있도록 지원하는 단말기로, 그 기능은 caller의 기능을 수행하는 UAC(User Agent Client)와 callee의 기능을 수행하는 UAS(User Agent Server)로 구분된다. 본 논문에서는 UAS 기능만을 갖춘 단말기에 한정하여 논한다.

SIP 서버는 Registrar Server, Proxy Server, Redirect Server로 구성된다. Registrar Server는 사용자의 현재 위치를 등록하고 사용자의 이동에 따른 위치 갱신 기능을 수행한다. Proxy Server(A)는 UAC로부터 콜 요청을 받아서 UAS의 위치를 알고 있는 Proxy Server(B)에게 콜 요청을 전달한다. Redirect Server는 UAC로부터 콜 요청을 받은 후 UAS의 Proxy Server(B) 주소를 UAC에게 알려주어 UAC가 다시 콜 요청을 수행하도록 하는 서버이다.

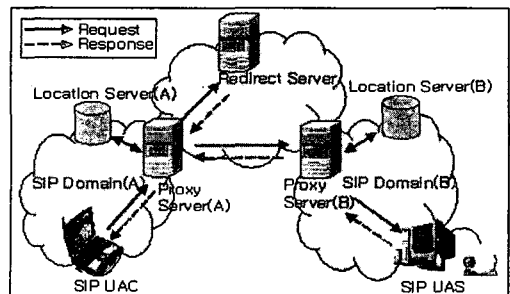


그림 1. SIP 시스템 구성

이 논문은 2004년도 중소기업 기술 혁신 기술 개발 사업의 위탁 과제로 연구비를 지원 받았음.

3. SIP UA 소프트웨어 설계

3.1 시스템 구조

VOVIDA사의 오픈 소스인 리눅스 기반의 SIPSET 1.5[4]와 MPEG4IP 1.0[5]을 이용한 SIP UA 구조는 그림 2와 같다.

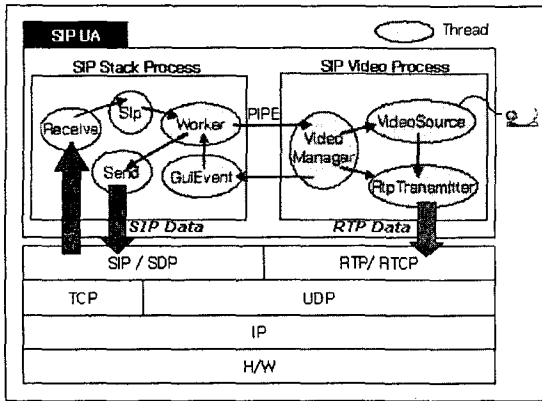


그림 2. SIP UA 시스템 구조

SIP UA는 현재 SIP의 스택 기능을 하는 프로세스와 SIP 비디오 프로세스로 나뉘어 프로세스 간에는 파이프(FIFO)를 이용하여 통신하고 쓰레드 간에는 큐를 이용하여 통신한다.

SIP 스택 프로세스는 콜 수신을 위해 SIP 메시지를 처리하는 프로세스이다. Receive 쓰레드는 caller인 UAC로부터 요청 메시지를 수신하며 Sip 쓰레드는 요청 메시지를 이벤트 형태로 변환한다. Worker 쓰레드는 이벤트 메시지를 처리하며 GuiEvent 쓰레드는 SIP 비디오 프로세스로부터 메시지를 수신하여 처리한다. Send 쓰레드는 SIP 응답 메시지를 UAC에게 보낸다.

SIP 비디오 프로세스는 콜이 설정된 후에 비디오 프레임 캡처하여 전송하는 프로세스이다. VideoManager 쓰레드는 콜 단계에 따라 파일을 통해 응답 메시지를 보내고 VideoSource 쓰레드와 RtpTransmitter 쓰레드를 제어한다. VideoSource 쓰레드는 카메라 드라이버로부터 비디오 프레임 캡처하고 인코딩한다. RtpTransmitter 쓰레드는 RTP[6] 패킷을 생성하여 caller인 UAC에게 RTP를 통해 비디오 데이터를 전송한다.

3.2 SIP 세션 설정

SIP 세션 설정 과정은 기본적인 SIP의 호출 흐름과 동일하며 다음과 같다.

- 1) caller인 UAC가 callee인 UAS에게 요청 메시지를 보내면 Receive 쓰레드는 요청 메시지를 수신하여 Sip 쓰레드 큐에 삽입한다.
- 2) Sip 쓰레드는 큐에 들어온 요청 메시지를 이벤트 메시지 형태로 변환하여 Worker 쓰레드 큐에 삽입한다.
- 3) Worker 쓰레드는 큐에 들어온 이벤트 메시지를 처리하여 파이프를 통해 VideoManager 쓰레드에게 메시지를 보낸다.
- 4) VideoManager 쓰레드는 비디오 프레임의 전송 유무에 따라 파이프를 통해 확인 메시지를 GuiEvent 쓰레드에게 보낸다.

- 5) GuiEvent 쓰레드는 파일을 통해 메시지를 받아서 Worker 쓰레드 큐에 삽입한다.
- 6) Worker 쓰레드는 큐에 들어온 메시지를 SIP 응답 메시지로 생성하여 Send 쓰레드 큐에 넣는다.
- 7) Send 쓰레드는 SIP 응답 메시지를 UAC에게 전송한다.
- 8) 단말 간 SIP 세션이 설정된다.

3.3 비디오 스트리밍

비디오 스트리밍은 SIP 세션이 설정된 후 시작되며 VideoSource 쓰레드는 비디오를 캡처하여 인코딩하는 쓰레드라고리즘은 그림 3과 같다.

```

/* 비디오 드라이버 버퍼 수는 2개로 가정한다 */
비디오 드라이버 버퍼 0번에 캡처를 지시한다;
비디오 드라이버 버퍼 1번에 캡처를 지시한다;

frameNo = 0; // 캡처 할 비디오 드라이버 버퍼 번호
total = 2;   // 비디오 드라이버 프레임 버퍼 수

{
    while(세션이 종료 될 때까지)
    {
        비디오 드라이버가 frameNo의 프레임을 캡처 완료 될 때 까지 기다린다;
        프레임을 버퍼로 복사한다;
        복사한 프레임을 H.261로 인코딩한다;
        인코딩 된 프레임을 RtpTransmitter 큐에 삽입한다;
        비디오 드라이버에게 frameNo에 프레임 캡처를 지시한다;
        /* 캡처 할 비디오 드라이버 버퍼 번호를 다음 버퍼 번호로 셋팅한다 */
        frameNo = (frameNo + 1) % total;
    }
}
    
```

그림 3. VideoSource 알고리즘에 대한 의사 코드

비디오 드라이버 버퍼는 더블 버퍼링을 하며 VideoSource 쓰레드는 캡처한 프레임을 처리하여 RtpTransmitter 큐에 메시지를 삽입하면 RtpTransmitter 쓰레드는 RTP 패킷을 생성하여 콜을 요청한 UAC에게 RTP를 통해 비디오 데이터를 전송한다.

4. 설계 이슈 및 성능 측정

4.1 사전 실험

본 연구팀은 임베디드 환경으로 포팅하기에 앞서 설계의 정확성과 성능을 평가하기 위해 사전 실험을 하였고 실험 환경은 표 1과 같다.

표 1. 실험 환경

Client	SIP UA (UAS)	Windows Messenger 4.7 (UAC)
OS	RedHat Linux 9.0	Windows XP
CPU	Intel Pentium4 1.8GHz	Intel Pentium4 2.8GHz
RAM	320MB	512MB
LAN	100Mbps	100Mbps
Camera	Logitech Pro 4000	없음

4.1.1 프레임 전송률

로지텍 카메라는 최대 초당 30프레임의 비디오 캡처를 지원한다. 본 실험의 목적은 초당 30프레임 캡처의 정확성을 검증하고

소프트웨어 측면에서 프레임 전송률의 제어 가능성을 확인한다. 또한 스케줄링에 따른 프레임 전송률의 변화 여부를 확인한다.

실험 결과 스케줄링에 의해 영향을 받지 않았으며 일괄적으로 초당 30프레임을 지원하였다. 또한 V4L(Video for Linux)[7] API를 이용하여 프레임 전송률의 제어가 가능하므로 QoS를 조절하는 하나의 방법이 된다.

#### 4.1.2 버퍼 크기

로직텍 카메라는 최대 640\*480의 해상도를 지원하며 현재 SIP UA는 Windows Messenger 4.7과 통신하기 위해 H.261로 인코딩한다. 한 프레임을 QCIF(176\*144)를 기준으로 VideoSource 쓰레드에서 요구되는 버퍼 크기를 측정 한 결과는 그림 4와 같다.

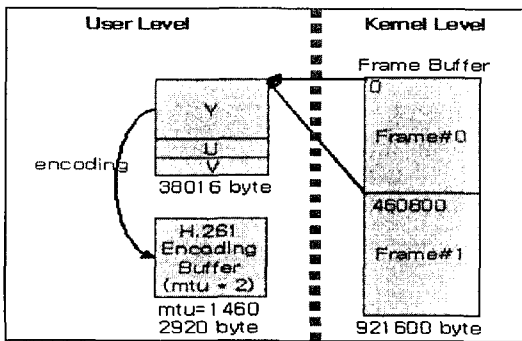


그림 4. VideoSource 에서 요구되는 버퍼 크기

실험 결과 총 요구되는 버퍼의 크기는 사용자 레벨에서 약 41Kbyte가 요구 된다. 이는 추후 임베디드 시스템으로 포팅 할 경우 메모리 크기가 중요시 되므로 각 해상도에 따른 정확한 버퍼 요구 사항의 측정이 필요하다.

#### 4.1.3 시간 측정 결과

비디오 캡처 시간, 인코딩 시간과 이를 전송하는데 걸리는 시간을 측정하여 실제 속도 차이로 인한 메모리 크기 증가와 비디오 전송 지연 문제 발생 여부를 검증하기 위해 한 프레임을 QCIF(176\*144)를 기준으로 측정해 본 결과는 표 2와 같다.

표 2. 시간 측정 결과

쓰레드	작업(한 프레임)	시간
VideoSource	비디오 캡처	33ms
	버퍼 복사+ 인코딩	0.9ms
RtpTransmitter	Rtp 패킷 전송	0.1ms

RtpTransmitter 쓰레드의 속도가 VideoSource 쓰레드 보다 상대적으로 빠르므로 지연 없이 비디오가 전송되며 두 개의 쓰레드 우선순위가 동일하기 때문에 메시지 큐 길이가 길어져 메모리의 증가나 지터의 문제는 발생하지 않는다.

#### 4.2 라이브 비디오 스트리밍 실험

그림 5는 실험환경을 바탕으로 왼쪽에는 UAC인 Windows Messenger 4.7 이며 오른쪽에는 UAS인 화상 카메라를 장착한 SIP UA의 모습이다. UAC가 UAS에게 콜을 요청하여 화상 카메라로부터 영상을 출력하는 모습을 나타내고 있다.



그림 5. 라이브 비디오 스트리밍 실험 모습

#### 5. 결론 및 향후 연구

본 논문에서는 SIP 시스템의 기본적인 동작을 설명하고, 기존의 VOVIDA사의 오픈소스인 SIPSET 1.5와 MPEG4IP 1.0을 이용하여 비디오 스트리밍을 지원하는 SIP 기반 단말기의 소프트웨어 구조에 대해 분석하여 이를 토대로 사전 실험을 하였다. 사전 실험을 통해 설계의 정확성을 검증하고 성능과 관련된 요소들을 파악 하였다.

향후 연구는 현재 리눅스 기반의 SIP UA를 임베디드 환경으로 포팅하여 QoS를 효율적으로 제어하는 방법을 개발하는 것이다.

#### 6. 참고문헌

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, IETF, June 2002
- [2] Yuan Zhang, "SIP-based VoIP network and its interworking with the PSTN", Electronics & Communication Engineering Journal, Vol. 14, No. 6, Pages: 273-282, Dec. 2002
- [3] Wook Hyun, MiYoung Huh, and ShinGak Kang, "An implementation of SIP servers for internet telephony", High Speed Networks and Multimedia Communications 5th IEEE International Conference on, Pages: 61-65, July 2002
- [4] VOVIDA SIPSET 1.5, <http://www.vovida.org>
- [5] MPEG4IP 1.0, <http://www.mpeg4ip.net>
- [6] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, IETF, July 2003
- [7] V4L(Video for Linux), <http://www.exploits.org/v4l>