
Wireless LAN 환경에서 임베디드 SIP User Agent 구현

박승환* · 이재흥**

An Implementation of Embedded SIP User Agent under Wireless LAN Area

Seung-Hwan Park* · Jae-Heung Lee**

요 약

본 논문은 무선의 임베디드 시스템 환경에서, VoIP 시스템을 구성하는 프로토콜 요소 중의 하나인 SIP를 이용한 User Agent의 구현에 관한 연구이다. User Agent는 설정 블록과, 주변 장치를 제어하기 위한 디바이스 쓰레드 블록, SIP 메시지를 처리하기 위한 SIP 스택 블록으로 구성하였다. 디바이스 쓰레드는 RTP 쓰레드 블록과 사운드 카드 처리 블록으로 구성하였으며, SIP 스택은 프락시 이벤트를 처리하는 워커 쓰레드 블록과 SIP 메시지를 전송하여 처리하는 SIP 트랜시버 및 SIP 쓰레드 블록으로 구성하였다. 하드웨어 플랫폼은 Intel XScale PXA255 프로세서 기반에 플래쉬 메모리, SDRAM, AC'97 오디오 코덱, 무선 랜카드와 연결된 PCMCIA 소켓이 내장된 보드를 구성하였으며, 오디오 입출력으로 마이크로폰과 헤드폰을 사용하였다. 본 연구의 실험을 위한 타겟 시스템 구성은 임베디드 리눅스 커널 2.4.19를 포팅하였다. 임베디드 시스템의 자원 효율을 높이고자, User Agent의 속성과 SIP 메소드의 기능을 최소화 하였고, TCP를 배제하여, 주변 장치 제어를 최소화함으로써, 자원의 소비를 12.9% 절감할 수 있었다.

ABSTRACT

This paper is about the research of the User Agent implementation under wireless embedded environment, using SIP which is one of protocol components construct the VoIP system. The User Agent is made of the User Agent configuration block, the device thread block to control devices and the SIP stack block to process SIP messages. The device thread consists of the RTP thread and the sound card device processing block. Furthermore, the SIP stack consist of the worker thread to process proxy events, the SIP transceiver and SIP thread to transfer and receive SIP messages. The H/W platform is a board included the Intel's XScale PXA255 processor, flash memory, SDRAM, Audio CODEC module and wireless LAN through PCMCIA socket, Furthermore a microphone and headphone is used by the audio I/O. The system has embedded linux kernel 2.4.19. For embedded environment, the function of User Agent and SIP method is diminished. Finally, the resource of system could be reduced about 12.9%, compared to overall system resource, by minimizing peripherals control and excepting TCP.

키워드

VoIP, SIP, Embedded, User Agent, Xscale

1. 서 론

기존의 PSTN(Public Switched Telephone Net-

work)망을 통해 이루어졌던 음성 서비스가 인터넷
이용자 수의 급속한 증가와 인터넷 보급이 대중화
되면서 IP(Internet Protocol)망을 통해 압축된 음성

신호를 전달하는 서비스로 변화하였다. 1995년 2월 이스라엘 VocalTec 사의 인터넷 폰을 시작으로 VoIP(Voice over Internet Protocol) 기술이 세상에 드러나게 되었고, 이후 실시간 통신을 위한 프로토콜과 이를 이용한 단말의 개발이 이루어져 왔다[1]. VoIP 프로토콜 중 H.323은 초기 PC-to-Phone형 인터넷 전화 시스템의 표준이 되었고, QoS의 단점을 극복하면서 안정화 되었다. 그러나 통화 세션 설정 과정이 복잡하여 네트워크 자원의 낭비를 가져올 뿐만 아니라, 단말의 프로그램이 소모하는 자원의 양도 증가한다는 문제가 대두되었고, 이를 해결하기 위해서 비교적 통화 세션이 간단한 텍스트 기반의 SIP(Session Initiation Protocol)가 등장하였다 [2]. 아울러, 네트워크 환경 역시도 휴대인터넷, 유비쿼터스 등의 모바일 환경과 부합하는 무선 통신 인프라 구축이 활발히 진행되면서, 무선 VoIP 폰 시장이 확대되고 있다.

본 논문에서는 무선 네트워크 개념과, 임베디드 환경으로 가져갈 수 있도록 Vovida의 SIP User Agent의 바이너리 파일 사이즈를 줄이는 연구를 시도하였다. SIP에 대한 개념과 이를 이용한 User Agent의 개념을 II장에 기술하였고, User Agent를 무선 임베디드 시스템 환경으로 가져가기 위한 타겟 보드의 설정과, User Agent 구현에 관하여 III장에 기술하였으며, 타겟 보드의 자원 사용을 최소화하기 위한 실험에 관하여 IV장에 기술하였다.

II. SIP와 User Agent

IETF의 MMUSIC WG (Multiparty Multimedia Session Control Working Group)에서 HTTP를 기반으로 개발된 SIP는 어플리케이션 레이어에서 세션의 생성, 해제 및 변경 기능을 수행하는 통화 세션 프로토콜로서, INVITE 메시지를 통하여 세션에 대한 정보를 전달하고, 참가자간의 호환 가능한 음성, 영상, 게임 및 문자와 같은 미디어 타입의 정보를 공유함으로써 세션을 생성한다[2].

User Agent는 SIP 메시지를 주고 받음으로써 클라이언트의 기능을 수행하거나, 서버의 기능을 수행하는 응용 프로그램이다. 세션을 정의하고, 참여자에 의해 세션이 성립이 되고, 세션을 공유하고, 종료하기까지 SIP 메시지를 받아서 큐에 저장하고, 파싱하고, 이벤트를 발생시켜, 이벤트에 따라 세션의 상태를 전이하게 된다. 이런 기능들을 수행하기 위해 몇 개의 쓰레드를 두어 보다 실시간으로 처리할 수 있도록 구성되어진다.

1. SIP 구조

그림 1은 SIP를 구성하는 프로토콜 구조이다. SDP(Session Description Protocol)는 SIP 멀티미디어 세션을 기술하는 프로토콜로서 미디어 전송 프로토콜이나, 포트, 음성 코덱 등이 포함된다. RTP (Real-time Transport Protocol)와 RTCP(RTP control Protocol)는 실시간으로 미디어 데이터를 전송하기 위해 사용되어진다. 하위 프로토콜로는 UDP (User Datagram Protocol)가 기본이나 TCP (Transmission Control Protocol)도 사용될 수 있다. 그림 2는 이러한 프로토콜이 시스템에 어떤 모습으로 참여하는지를 보여준다.

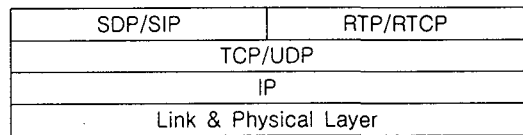


그림 1. SIP 프로토콜 구조

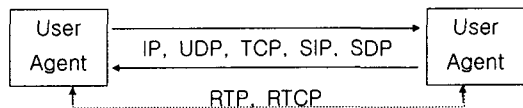


그림 2. SIP 보조 프로토콜 상호 운영

2. SIP 구성

SIP 구성요소는 그림 3 시스템 구성도와 같이 User Agent, 프락시 서버, 리다이렉트 서버로 구성된다. User Agent는 SIP 요청(request) 메시지를 전송하기도 하고, 받은 SIP 요청 메시지에 대한 내부적 판단에 의하여 SIP 응답(response) 메시지를 전송하기도 한다. 프락시 서버는 User Agent로부터 받은 요청 메시지를 다른 프락시 서버로 직접 전송하거나, 하나의 요청 메시지를 여러 서버로 동시에 전송할 수 있다. 리다이렉트 서버는 사용자에게 요청 메시지의 목적지 URL을 알려주어 사용자가 직접 전송하도록 한다.

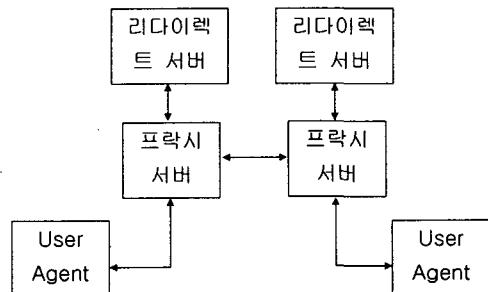


그림 3. 시스템 구성도

3. SIP 주소

SIP 주소 형식은 이-메일 주소와 유사한 "User_name@host_name"의 형식을 갖는다. "User_name"은 사용자를 식별할 수 있는 ID이며, "host_name"은 도메인 명이나 IP주소가 가능하다.

4. SIP 메시지

그림 4와 같이 SIP 메시지는 요청 메시지와 응답 메시지 두 종류가 있는데, 이 메시지들은 통신의 자세한 부분을 기술하는 여러 가지 헤더를 포함하고, status 라인에 의해 결정된다. 요청 메시지는 프로토콜 버전에 따르는 메소드에 의해서 시작되고, 응답 메시지는 프로토콜 버전과 요청 메시지에 대한 성공 여부로 시작한다. via로 시작되는 헤더 영역은 호출자, 피호출자, 메시지 경로, 메시지 본체의 타입과 길이 등을 나열하기 위해 사용하고, 미디어 세션에 대한 상세한 기술은 Body영역에 기술되며, SDP의 버전, 세션 주체자에 대한 정보, 미디어 채널의 수신대기 구분, 세션의 시작 시간, 미디어 스트림 형식을 나열하기 위해 사용한다[3].

	Request	Response
Status	method URL SIP/2.0	SIP/2.0 status reason
Header	Via : SIP/2.0/UDP host:port From : User <sip:from_User@source> To : User <sip:to_User@destinamtion> Call-ID : localid@host Cseq : seq#method Content-Type : media type of body Content-Length : length of body	
	CRLF	
Body	v=0 o=origin_User timestamp IN IP4 host c=IN IP4 media destination address t=0 0 m=media type port RTP payload types	

그림 4. SIP 메시지 구조

5. User Agent

User Agent는 가장 핵심이 되는 state machine과 SIP 메시지를 처리하는 블록, 주변 장치를 제어하는 블록으로 구성된다[5]. state machine은 그림 5에서와 같이 쓰래드에서 발생하는 이벤트를 파싱하여 이벤트에 따라 Idle, Dialing, Trying, Calling Error, Ringing, InCall 상태로 전이한다. Idle 상태에서 세션 설정 요청을 하기 위하여, 피호출자의 SIP 주소를 입력하게 되면, Dialing 상태로 전이하고, 목적지 SIP주소를 파싱하여 주소에 문제가 있으면 Error 상태를 거쳐 Idle 상태로 전이하고, 그

렇지 않을 경우 피호출자의 상태를 판단하는 Trying 상태로 전이한다. 이때, 피호출자의 상태가 이미 다른 세션이 이루어져 있거나, SIP의 메시지에 기술된 미디어 타입을 주변 장치가 처리 할 수 없는 경우 Error 상태를 거쳐 Idle 상태로 전이한다. 판단한 결과에 문제가 없다면, 세션을 정의하고, 응답 메시지를 받아 세션을 성립하게 된다.

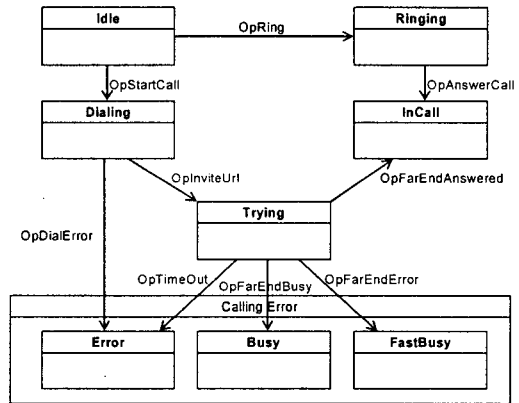


그림 5. User Agent 구성 쓰래드

III. User Agent 구현

User Agent 구성 쓰래드는 그림 6과 같이 크게 디바이스 쓰래드, RTP 쓰래드, 프락시 쓰래드로 구성된다. 디바이스 쓰래드는 미디어 타입에 따라 주변 장치를 제어하게 되는데, 음성 포맷의 데이터를 처리하기 위하여 사운드 카드를 제어하도록 하였으며, 텍스트, 이미지등의 미디어 타입을 갖는 세션에도 참여할 수 있도록 하였다. RTP 쓰래드는 미디어 데이터의 실시간 전송 기능을 수행하도록 구현하였다. 프락시 서버는 User Agent 속성을 설정하여 User Agent 개체를 만들어내는 UaBuilder

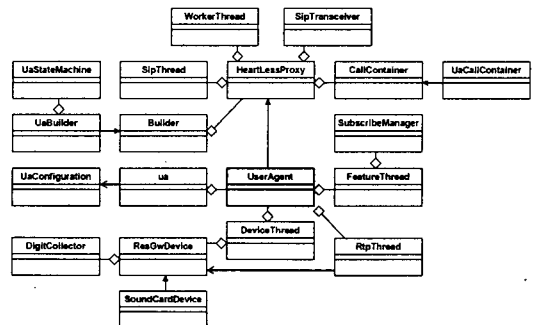


그림 6. User Agent 구성 쓰래드

와 SIP 메시지를 처리하는 쓰레드들로 구성하였다. Worker thread는 프락시 이벤트가 있을 때 마다 SIP 메시지를 저장하는 기능을 하고, SIP 트랜시버는 통화 세션에서 SIP 메시지를 전송하는 역할을 하고, SIP 쓰레드는 SIP 스택으로부터 받아온 SIP 메시지를 어플리케이션으로 전송하는 기능을 한다.

IV. 실험 결과

타겟 보드의 H/W 는 그림 7과 같이 ARM 코어와 다양한 주변장치를 온칩화한 Intel XScale PXA255 프로세서와 32MByte의 플래쉬 메모리, 128MByte의 SDRAM, AC '97 CODEC, Full Uart, PCMCIA socket, Wireless LAN으로 구성하였다 [6].

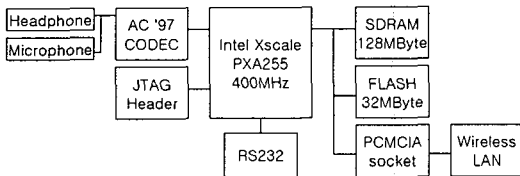


그림 7. 타겟 보드 H/W 구성

타겟 보드에는 임베디드 리눅스 커널 2.4.19를 ARM-PXA255 아키텍처에 대한 패치를 하여 포팅하였고, 오디오 CODEC과 PCMCIA, Wireless LAN 드라이버를 커널에 탑재하였으며, gcc-3.3을 ARM용으로 컴파일한 크로스 컴파일러를 사용하여, c++로 구현한 User Agent를 컴파일 하였다[7]~[9]. 크로스 컴파일 된 User Agent의 바이너리 파일을 NFS로 마운트 하여 PC환경의 User Agent와 음성 통화를 테스트함으로써 무선 환경에서 임베디드 SIP User Agent의 실행 가능성을 확인하였고, 80 Mbyte의 램디스크 안에 직접 바이너리 파일을 실어 독립적인 User Agent의 기능이 가능하였다.

임베디드 환경으로 가져가기 위한 노력으로 Invite 메시지를 보낼 때, 리다이렉트 서버를 가상으로 처리하기 위해, 콘솔 상태에서 수신자의 주소를 써주는 대신 단축다이얼 형식의 설정을 하였으며, 자원 소비의 최소화를 위하여 VoIP 시스템의 최소한의 기능만을 수행하도록 하였다. User Agent의 속성과 SIP 메시지의 기능을 최소화하고, TCP를 배제하였으며, 주변장치로는 사운드 카드만을 제어하도록 구현한 결과, 68 Mbyte의 용량을 갖는

Vovida의 SIP User Agent와 비교하여 57 Mbyte의 용량으로 줄임으로써, 12.9%정도의 메모리를 절약할 수 있었다. 그 외에 디버깅을 위한 방대한 소스들을 제거한다면 자원의 소비를 더 줄일 수 있을 것으로 기대된다.

아울러, SIP-MGCP, SIP-H323과 같은 SIP 메시지를 변환 해 주는 프락시 서버의 기능을 확장할 수 있을 것으로 보이며, SIP가 1개 이상의 다양한 미디어 타입의 데이터 처리를 지원하기 때문에, 영상 데이터를 처리할 수 있는 환경을 구축하여, 화상회의와 같은 동영상 서비스로도 확장해 나갈 계획이다.

V. 결 론

본 논문은 통화 세션 설정 과정이 비교적 간단한 SIP를 이용하여 무선 인터넷 환경의 임베디드 SIP User Agent의 구현에 관한 연구이다. 현재 PDA 시스템에 사용되고 있는 Intel XScale PXA-255 프로세서에 임베디드 리눅스를 포팅하고, wireless LAN과 사운드 카드 장치에 대한 드라이버를 적재하여, 타겟 보드의 환경을 구축하였고, 자원의 소비를 최소화 하여 SIP User Agent를 구현한 결과 음성 통화의 가능성을 확인 하였다. 자원의 소비를 더 줄이기 위한 노력과, 다양한 미디어 타입의 데이터를 처리할 수 있는 환경을 위해 확장해 나갈 계획이다.

참고문헌

- [1] 박성진, "분산형 SIP 기반 PC to Phone 시스템의 설계 및 구현", 공학석사 학위논문, 2002.
- [2] Vovida, "Voice over IP Protocols An Overview".
- [3] '주간 기술 동향', 한국전자통신연구원, 제 1056호, p.14-29, 2002.
- [4] J. Rosenberg et al. "SIP : Session Initiation Protocol", RFC 3261, June 2002.
- [5] Luan Dang, Gullen Jennings & David Kelly, "Practical VoIP using Vocal", July 2002.
- [6] Intel, "Intel PXA255 Processor developer's manual".
- [7] Steve Furber, "ARM System-on-chip Architecture", p.49, ADDISON-WESLEY.
- [8] 이명근, 이상정, 조성범, 임재용, "실시간 처리 리눅스 기반 VoIP 시스템 설계 및 구현", 정보과학회 발표문집(A), 제 28권 제 1호, p.251-

289. 2001.

[9] Alessandro Rubini, "Linux Device Drivers",
O'REILLY.

저자소개

박승환(Seung-hwan Park)



e-mail : shpark@artoa.hnu.ac.kr
2003년 한밭대학교 컴퓨터공학과 공
학사
2005년 한밭대학교 정보통신 전문대
학원 컴퓨터공학과 공학석사
※관심분야 : 임베디드 시스템, VoIP,
커널



이재흥(Jae-heung Lee)

e-mail : jhlee@hanbat.ac.kr
1983년 한양대학교 전자공학과 공학
사
1985년 한양대학교 대학원 전자공학
과 공학석사
1994년 한양대학교 대학원 전자공학
과 공학박사
1989년~현재 한밭대학교 정보통신·컴퓨터공학부 교
수
※관심분야 : CAD 및 VLSI 설계 임베디드 시스템,
SoC