

# 7가지 상태를 이용한 SIP 인터넷 전화연결 시스템 설계 및 구현

(Design and Implementation of SIP Internet Call-setup  
System using Seven States)

신 용 경 \*      김 상 욱 \*\*

(YongKyoung Shin)      (SangWook Kim)

**요 약** SIP은 IP 전화 통신에서 통신 셋업을 위해 사용되는 주요 프로토콜 중 하나이다. SIP을 통한 통신은 사용자 요구에 따라 다양한 상태가 필요하다. 본 논문은 기존의 SIP 프로토콜을 이용하여 전화연결 시 사용자의 요구에 따라 수시로 변하는 상태를 7가지로 정의하고, 객체 지향적인 시스템 설계에 따라 각 상태를 전이하는 이벤트를 두어 개발자 관점에서 SIP 프로토콜을 이용한 새로운 응용서비스를 개발하고자 할 때 객체지향적인 시스템 설계를 할 수 있다. 상용망에서 RFC 3261에서 제시하는 Call-Setup과정을 따라 인터넷 전화연결 시스템을 설계하면 예외처리 및 과도한 트래픽이 발생하여 시스템 오류를 불러올 수 있다. 이 경우에 시스템의 State를 보고 예상된 이벤트일 경우 정형화된 처리루틴으로 대처하고, 그렇지 않을 경우도 예외상황을 효율적으로 처리할 수 있다. 이벤트 처리루틴은 FSM으로 설계 및 구현하였다.

키워드 : 상태, SIP, 전화연결, FSM

**Abstract** The Session Initiation Protocol (SIP) is one of the major protocols used in call-setup over IP telephony. The SIP-sigaled calls use many-sided states according to a request of user. In this paper, we suggest seven states and some events that help developers to design and implement new applications efficiently. And they enable an object-oriented design of the system. If you design the call-setup procedure only by the processing model suggested in RFC 3261 over commercial network, a fatal error may occur in the system because of heavy data traffic or unpredicted exception cases. However, according to the suggested seven states, if they are predefined events in the current system state, the standardized processing routine is executed. Otherwise, they can be processed by the exception routine in system. All event processing routines are designed and implemented using Finite State Machine (FSM).

**Key words** : State, SIP, Call-Setup, FSM

## 1. 서 론

IP 텔레포니 기술이 발전하면서 SIP 프로토콜을 이용한 전화연결 서비스가 보편화되고 있다[1]. 현재 상용화되고 있는 인터넷 전화에 이용되는 프로토콜로는 SIP와 H.323이 있는데 SIP가 text를 기반으로 하여 H.323보다 복잡도가 낮고, 확장성이 높아 최근에는 SIP를 선호하는 추세이며[2,3], 따라서 다양한 서비스를 개발하기 위한 SIP구성요소의 확보가 중요하다[2]. RFC 3261에서

제시하는 SIP 프로토콜은 security문제뿐만 아니라 상용화 수준으로 개발하기는 많은 구성요소들이 보완되어야 하며, 현재도 RFC 3311("SIP UPDATE Method") 이나 RFC 3313("Private SIP Extensions for Media Authorization") 등과 같이 SIP 프로토콜의 구성요소를 보완하는 작업을 IETF에서 계속해서 진행 중에 있다 [4-6]. 이와 같이 SIP 프로토콜이 상용 망에서 많이 이용되지만 아직까지 문제점들을 많이 드러내고 있다고 볼 수 있다[7,8]. SIP 프로토콜을 이용하여 상대방에게 전화를 걸거나, 상대방으로부터 전화가 걸려오는 경우 또한 여러 가지 예외상황이 발생할 수 있다. 과 트래픽 전송문제, 망 오버헤드 문제, 패킷로스 및 재전송 등의 예상치 못한 문제에 대해서 일일이 기술하여 캐치하기는 개발자들에게는 어렵다[9,10]. 또한, 이와 관련된 대

\* 정 회 원 : LG전자 SW Center 연구원  
lemonsage@lge.com

\*\* 정 회 원 : 경북대학교 전자전기컴퓨터학부 교수  
swkim@cs.knu.ac.kr

논문접수 : 2007년 2월 15일

심사완료 : 2007년 9월 10일

처방안을 제시하고 있는 RFC문서는 존재하지 않는다.

본 논문에서는 사용자가 전화를 걸거나 받을 때 7가지 State를 기반으로 Event가 발생하면 그에 적합한 수식을 적용하고 어떻게 효율적으로 전이하는지를 보인다.

본 장에 이어서 제2장에서는 SIP프로토콜 관련 연구를 소개하고, 제3장에서는 SIP인터넷 전화연결 시스템을 설명하고, 제4장에서는 7가지 state를 보이고, 각 state를 분류하는 기준 및 Event가 들어오면 state가 어떻게 전이하는지를 설명한다. 제5장에서는 위의 내용을 시스템에 적용하여 검증한 결과를 보이고, 제6장에서는 결론 및 향후 연구 계획을 기술하고자 한다.

## 2. 관련 연구

### 2.1 RFC 3261에서 제시하는 전화연결과정

SIP표준문서는 INVITE/ACK/OPTIONS/BYE/CANCEL/REGISTER의 Request Method를 이용하여 UAC (User Agent Client)들간의 Call-Setup이 이루어진다 [1]. 그림 1은 Alice와 Bob간의 전화연결 과정을 보인다. 그림 1과 같이 [1]에서 제시하는 SIP Call-Setup은 단순히 'INVITE' Method를 만들어 Alice가 등록되어 있는 Proxy 서버(Atlanta.com)에 메시지를 보내고 Bob도 Proxy 서버(Biloxi.com)로부터 'INVITE' Method를 받아 처리하여 Response를 보내는 일련의 과정을 수행한다. 그러나 proxy서버의 상태나 Bob의 상태에 따라 발생하는 예외상황에 대한 처리과정을 구체적으로 제시하지 않는다. 이동중인 Alice가 Atlanta.com의 proxy서

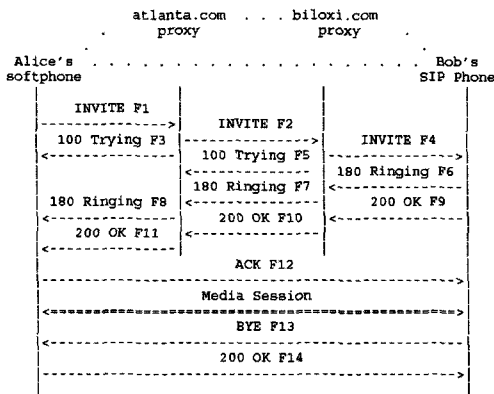


Figure 1: SIP session setup example with SIP trapezoid

```

INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhs
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
    
```

(Alice's SDP not shown)

그림 1 RFC 3261에서 제시하는 SIP 전화연결 과정

버에 'INVITE'를 보내고 나서 100 Trying을 받기도 전에 다른 proxy서버로 이동하였거나 이동중인 경우 Alice는 'INVITE'를 새로 보낼 것인지 이동한 proxy서버로부터 100 Trying을 기다릴 것인지 상태를 관리할 필요가 있다.

### 2.2 The Vovida Software

www.vovida.org에서 공개된 SIP프로토콜을 볼 수 있다[11]. 여기서 제시하는 SIP프로토콜의 구조는 RFC2543의 functionalities를 만족시키고 Methods는 'INVITE', ACK, BYE, CANCEL, REGISTER, OPTIONS, INFO로 구성되어 있다. 그림 2는 SIP프로토콜을 이용하여 VoIP를 하는 과정을 보인다[12]. 이때 사용되는 State는 INACTIVE, INCOMING\_REQUEST, CALL\_IN\_PROGRESS, FATAL\_ERROR로 4가지만을 제시하고, 자신의 단말뿐만 아니라 상대 단말의 예외상황 및 performance를 고려하지 않는다. VoIP와 같은 음성데이터일 경우 치명적인 예외상황이 발생하지 않겠지만 대용량 데이터나 보안을 요하는 중요 데이터 전송시 이러한 SIP signal구조는 서비스 동안에 치명적인 문제를 일으킬 수 있다.

## 3. SIP인터넷 전화연결 시스템 구조

본 논문에서 제시하는 Seven States를 설명하기 이전에 적용되는 시스템의 구조를 우선 이해하여야 한다. 그림 3은 Seven States를 이용한 SIP인터넷 전화연결 시스템 구조를 보인다. 크게 3개의 모듈로 구분하며, UI와 다른 Task와의 웬들링을 담당하면서 하부 모듈에서 발생하는 이벤트를 받아 처리하는 Service Main Manager (이하 SM이라 함)가 있고, 전화연결을 위하여 SIP 프로토콜을 encoding/decoding하는 프로토콜 관련 Stack이 필요하다. 이를 관리하는 Protocol Stack Manager (이하 PM이라 함)가 있다. 마지막으로 전화연결 프로토콜을 이용하여 통신이 이루어지면 음성이나 영상과 같은 미디어 전송이 필요하다. 이를 Media Stack Manager(이하 MM이라 함)에서 관리한다.

### 3.1 Service Main Manager(SM)

SM은 UI와 다른 System task들과의 Interface역할을 하며 SIP call과 연관된 이벤트를 생성 및 처리하고 사용자 정보관련 DB를 관리한다. Service Event Process와 Presence DB Manager, Synchronization Controller, User Profile Configuration Manager, NVDB (Non-Volatile DB) Manager와 Audio/Video Controller를 가지고 있어 각 DB와 Multimedia Codec와 연계하여 사용자 요구사항을 만족시킨다.

- Service Event Process

UI에서 들어오는 정의된 이벤트를 해석하여 다른 모

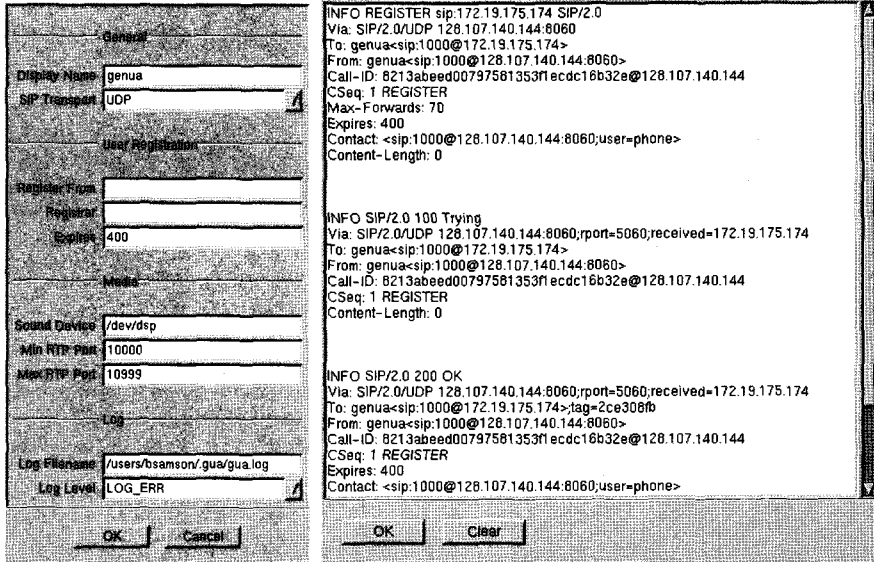


그림 2 Vovida.org SIP 전화연결과정

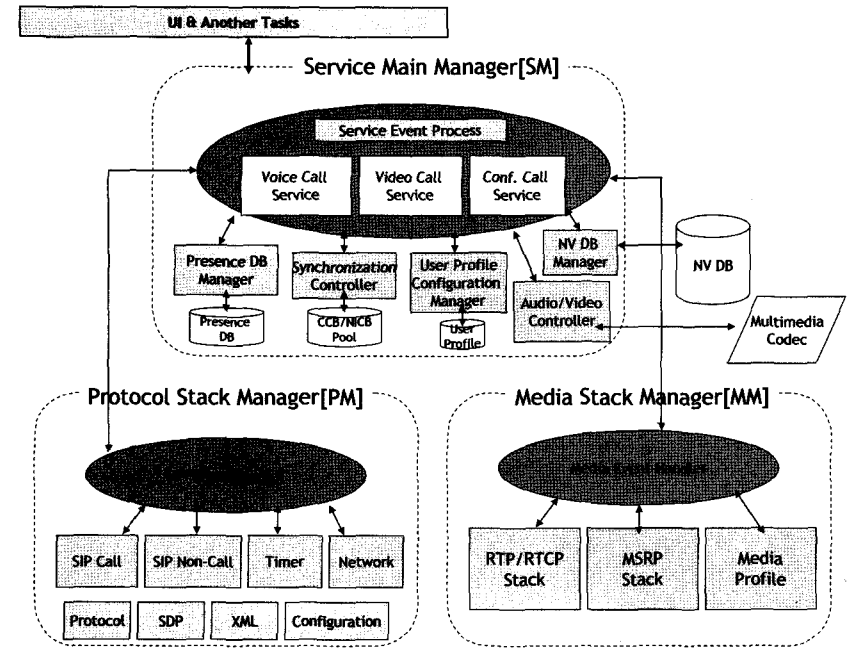


그림 3 SIP 인터넷 전화연결 시스템 구조

들에 전달하고, 다른 모듈에서 들어오는 이벤트를 해석하여 UI로 결과를 알린다. SIP로 전화 연결하여 원하는 서비스를 Application이라 하면 Service Event Process는 어떤 Application을 위한 SIP 전화 연결인지를 구분하여 관리한다.

- Presence DB Manager

Buddy의 Presence 및 Group 정보를 저장 관리한다. Buddy의 Presence 정보로는 통신 상태(Status), 그룹 정보, 이름, 성별, 기호 등이 있으며, Application의 요구 사항에 따라 이곳에 Presence정보를 추가할 수 있다. Buddy의 Group 정보로는 Group 이름, Group ID 등이 있으며, 마찬가지로 Application의 요구 사항에 따라

이곳에 Group 정보를 추가할 수 있다. DB형태로 저장되는 Buddy DB, Group DB를 관리할 수 있는 Function 들로 구성되어 있다. 또한 DB의 Access, Delete, Add, Update를 위해 Linked-List Mechanism을 응용하여 적용한다.

- Synchronization Controller

Overlap Transaction, Mid-Call Transaction을 처리한다. 동시에 여러 개의 Transaction 또는 Dialog를 구별하고, 동시에 처리되는 각각의 Transaction 또는 Dialog들 상호간의 독립성을 최대한 유지하는 역할을 한다.

- User Profile Configuration Manager

Application 별로 Provisioning Profile Data를 저장, 초기화, 관리한다. 예를 들면 Authentication Parameter, Registration(Subscription) Expire Value를 관리한다. PM에서 이 Profile Data를 이용하여 Call Object의 초기 Data를 구성한다.

- NVDB Manager

Server URI/IP Address, Service Enable/Disable Parameter, Receiving Mode, Login Process 등을 결정하는 Non-Volatile Data를 관리한다.

- Audio/Video Controller

멀티미디어 코덱을 관리한다.

### 3.2 Protocol Stack Manager (PM)

Protocol Stack Manager내에서 SIP Dialog또는 SIP Call Procession을 위해 사용되는 Call Object를 관리한다. RFC 3261에서 제시하는 SIP기본 mechanism을 사용하고, SM에서 전달받은 사용자 정보를 기본으로 하여 Call을 생성한다. Network Manager가 있어 PPP Connection 설정, 해제, 상태체크 등을 수행하고 SIP를 위한 Socket Control 및 SIP Message Sending, Receiving을 수행한다. SDP Manager에서 Media 전송 및 수신을 위하여 Message Forming 및 Negotiation을 수행한다.

### 3.3 Media Stack Manager (MM)

SIP Call 연결이 설정되면 두 단말은 데이터를 주고받는다. 이때 데이터로는 음성, 이미지, 동영상 등 다양한 형태를 지원한다. SIP 프로토콜로는 Call연결만 가능하고 데이터를 위한 채널은 다른 프로토콜을 이용하여야 한다. MM은 데이터를 주고받기 위한 추가적인 프로토콜을 관리한다. 본 논문에서 제시하는 데이터 교환 프로토콜로는 MSRP(Message Session Relay Protocol)[13]를 사용한다. Call 연결시 SDP[1]로 데이터 교환을 위하여 어떤 프로토콜을 이용할 것인지 정의하고 포트번호를 negotiation한다. 서로간의 네트워크 채널이 정상적으로 열리면 MSRP 형식으로 데이터를 만들어 전달

하고, 이벤트 및 전송상태를 MM에서 체크한다. SIP Call이 해제되면 MSRP채널도 자동으로 닫혀야 하므로 SM에서 실시간으로 이벤트를 주고받을 수 있다.

## 4. Seven States를 이용한 전화연결

본 논문은 Seven States를 두어 전화를 거는 단말(Mobile Originator, 이하 MO라 함)과 전화를 받는 단말(Mobile Terminator, 이하 MT라 함)에서 발생할 수 있는 예외상황을 효율적으로 처리하는 과정을 설명한다. RFC3261에서 제시하는 Proxy Server나 Relay Server에 의해 나타날 수 있는 MO와 MT의 예외상황을 제외하고는 Server 시스템 내부 예외상황에 관련된 내용은 본 논문의 범위를 벗어나므로 다루지 않는 것으로 한다.

### 4.1 일반적인 전화연결 과정

MO는 REGISTER과정을 통하여 Proxy Server에 주소 및 Session정보를 등록한다. 등록과정이 성공적으로 끝나면 사용자는 연결하고자 하는 상대방을 선택하여 전화연결을 시도하는데 MO는 현재의 상태정보를 가지고 'INVITE' Method를 text기반으로 encoding하여 proxy서버에 보내면 상대방의 상태에 따라 '100 Trying'이 오거나 '200 OK' response message가 온다. 같이 전달되는 SDP message를 해석하여 Call-Setup이 성공하였는지 실패하였는지 아니면 내가 가지고 있는 상대방의 미디어정보 및 주소정보를 변경해야 하는지를 판단하게 된다. SDP정보를 해석하여 적합한 미디어 채널을 열어 음성이나 영상을 주고받는다. 전화연결을 끊을 경우는 MO이든 MT이든 관계없이 한쪽에서 'BYE' request message를 생성하여 보내고 '200 OK' response message를 받으면 미디어 세션 및 SIP 인터넷 전화연결이 끊어진다.

MT도 REGISTER과정이 우선되어야 하며, 등록과정이 성공적으로 끝나면 상대방으로부터 전화가 걸려온다. 상대방이 연결 시도하는 채널의 형태가 음성인지 영상인지에 따라 MT는 encoding 및 decoding가능 여부를 결정해야 한다. [1]에 따르면 MO가 실어 보내는 SDP의 session정보를 해석하여 미디어 채널 생성이 가능한지를 판단하여 '200 OK' response message를 보내고 그렇지 않을 경우 'Request Failure 4xx', 'Sever Failure 5xx', 'Global Failure 6xx'중에서 MT의 상태를 알릴 수 있는 적절한 Response message를 선택하여 보낸다. '200 OK'를 보낸 경우 MO가 잘 받았는지 확인하기 위하여 'ACK' request message를 기다린다. 'ACK'를 받으면 미디어를 위한 채널을 열어 데이터를 주고받는다.

### 4.2 SIP전화연결 시 발생하는 이벤트

본 논문에서 이벤트 naming rule은 다음과 같이 정의한다.

(대분류)\_(중분류)\_(접미사)

- (1) (대분류) : 이벤트를 처리하는 주체 M|PM|MM
    - SM : 그림 3에서의 Main Manager
    - PM : 그림 3에서의 SIP Stack Manager
    - MM : 그림 3에서의 Media Stack Manager
  - (2) (중분류) : 의미 전달이 명확하도록 이벤트의 Operation에 대한 동사나 명사
  - (3) (접미사)
    - 반드시 짝을 이루어야 하는 경우에 대한 이벤트 postfix
      - i. REQ(ReQuEst): Upper Layer에서 Lower Layer로 특정 Operation을 요청
      - ii. RES(ReSpOnse): REQ에 대한 Operation을 완료한 이후에 대한 응답 보냄
      - iii. RPT(RePorT): Lower Layer에서 Upper Layer로 특정 Operation을 요청
      - iv. CNF(CoNFirm): RPT에 대한 처리가 완료되었음을 Lower Layer로 다시 알려주는 응답을 보냄
    - 짝을 이루지 않고 홀로 존재하는 이벤트 postfix
      - i. IND(Indication): Lower Layer에서 Upper Layer로 특정 상황을 알리고, 이에 대한 추가 이벤트 primitive 교환이 필요하지 않음
      - ii. CMD(Command): Upper Layer에서 Lower Layer로 특정 상황을 알리고, 이에 대한 추가 이벤트 primitive 교환이 필요하지 않음
- 위의 naming rule에 따라 SIP인터넷 전화연결 시 발

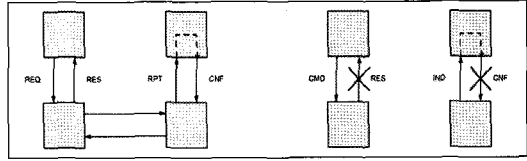


그림 4 이벤트 진행 방향에 따른 naming rule

생하는 이벤트를 표 1에서 보인다. 표 1에서는 본 논문에서 제시하는 이벤트 이름과 어떤 상황에서 이벤트가 발생하는지 설명하고 발생 방향은 화살표 왼쪽이 Sender가 되고 오른쪽이 Receiver가 되어 해당 메시지를 주고받는다. 4.3 Seven States에서 제시하는 각 State의 전이는 다음의 이벤트를 기본으로 설명하고 있다. 발생방향에서 UI는 User Interface를 나타낸다. 이벤트 설명에서 발신자는 전화를 거는 사람, 수신자는 전화를 받는 사람, 발신 측은 전화를 거는 단말(MO), 수신 측은 전화를 받는 단말(MT), 상대방은 발신 측 혹은 수신 측을 말한다.

4.3 Seven States

본 논문에서 제시하는 Seven States는 NULL, CLEARING, ESTABLISHING, WAITMEDIAANSWER, WAITUSERANSWER, RESPONDING, ACTIVE이다.

4.3.1 NULL

MO와 MT가 Proxy Server에 등록을 거치고 유효한 사용자 인증을 받은 후 초기화 되는 state이다. 단말은 NULL상태에서 'INVITE' request를 보내는 쪽이 MO가 되고 받는 쪽이 MT가 된다.

표 1 이벤트

Event Name	Event Contents	Direction
SM_ESTABLISH_CMD	발신자 요구에 따라 SIP전화연결 요청이 시작되는 이벤트	UI→SM
SM_END_CMD	발신자가 전화연결을 해제하는 이벤트, 전화연결시도 중에서 이벤트 발생이 가능함.	UI→SM
SM_INCOMING_RPT	수신자에게 전화가 왔음을 알리는 이벤트	SM→UI
SM_INCOMING_CNF	수신자가 걸려온 전화에 승인 응답을 주는 이벤트	UI→SM
SM_RELEASED_IND	상대방에 의해 전화연결이 해제되었음을 User에게 알리는 이벤트	SM→UI
PM_SENDINVITE_CMD	'INVITE' message를 만들어 수신 측에 전달하도록 요청하는 이벤트	SM→PM
PM_FINALRESPONSE_IND	수신 측으로부터 'INVITE'에 대한 response message가 왔음을 알리는 이벤트	PM→SM
PM_END_REQ	BYE message를 만들어 상대방에게 전달하도록 요청하는 이벤트	SM→PM
PM_END_RES	수신 측으로부터 BYE에 대한 response message가 왔음을 알리는 이벤트	PM→SM
PM_ACK_CMD	'200 OK' response를 수신하였음을 알리기 위하여 ACK message를 만들어 수신 측에 전달하도록 요청하는 이벤트	SM→PM
PM_ACK_IND	발신 측에서 ACK message가 왔음을 알리는 이벤트	PM→SM
PM_RECEIVEINVITE_IND	발신 측에서 'INVITE' message가 왔음을 알리는 이벤트	PM→SM
PM_FINALRESPONSE_CMD	'INVITE' message를 수신하고 response를 상대방에게 전달하도록 요청하는 이벤트	SM→PM
PM_END_RPT	발신 측에서 BYE message가 왔음을 알리는 이벤트	PM→SM
PM_END_CNF	BYE message에 대한 response를 상대방에게 전달하도록 요청하는 이벤트	SM→PM
PM_SUBSCRIP_CMD	서버에게 SUBSCRIBE message를 보내도록 요청하는 이벤트	SM→PM
MM_SESSIONCREATE_REQ	Multimedia Data를 받기 위하여 Session을 생성하도록 요청하는 이벤트	SM→MM
MM_SESSIONCREATE_RES	Session 생성결과를 알리는 이벤트	MM→SM
MM_SESSIONDEL_CMD	Multimedia Data를 받거나 주기 위하여 생성한 Session을 제거하도록 요청하는 이벤트	SM→MM

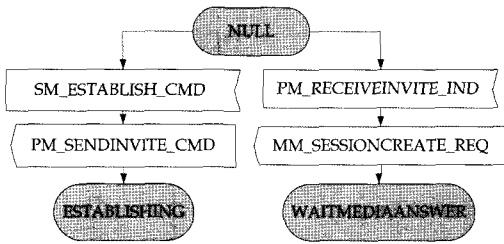


그림 5 NULL state의 전이 과정

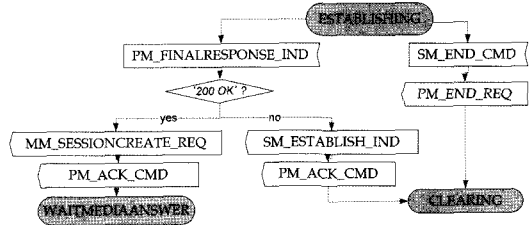


그림 7 ESTABLISHING state의 전이과정

4.3.2 CLEARING

RFC 3261에서 제시하는 SIP Signaling은 '200 OK'를 보내고 나서도 'ACK'를 받아야지만 최종적으로 미디어 세션이 열린다. '200 OK'가 아닌 시스템상의 문제로 연결이 실패할 경우라도 Failure message를 보내고 'ACK'를 받아야 한다. 그렇지 않을 경우 재전송 해야 한다. 따라서 전화연결이 실패할 경우 이전 state에서 NULL상태로 전이하게 되면 'ACK'를 받았는지 확인 할 수 없으므로 상대방의 시스템이나 서버에 치명적인 error를 일으킬 수 있다. 따라서 CLEARING 상태로 우선 전이 한 다음 NULL상태로 전이하기 위한 처리를 반드시 해야 한다.

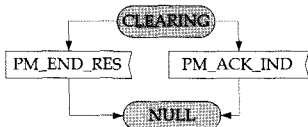


그림 6 CLEARING state의 전이과정

4.3.3 ESTABLISHING

MO가 상대방에게 전화를 걸어 연결의사를 묻고 있는 상태를 말한다. MO의 미디어 정보를 SDP에 실어 SIP Signal과 함께 보내면 MT로부터 연결 가능여부를 알려 준다. MO는 ESTABLISHING 상태에서 MT의 결과값을 해석하여 다른 State로 전이한다.

4.3.4 WAITMEDIAANSWER

상대단말에서 요구하는 미디어 채널의 capacity나 코덱 정보 및 performance로 내 단말과 통신가능한지를 판단하는 상태를 말한다. MO는 '200 OK' response message와 함께 전달되고, MT는 'INVITE' request message에 실려 온다. 따라서 MO는 ESTABLISHING 상태에서 '200 OK'가 오면 WAITMEDIAANSWER로 전이하고 그렇지 않은 경우 CLEARING으로 전이한다.

4.3.5 WAITUSERANSWER

MT는 상대방에서 요구하는 미디어 정보로 미디어 채널 연결이 가능하더라도 최종사용자(End-User)에게 전화연결을 할 것인지 의사결정을 요구해야 한다. 최종사용자가 상대방의 전화를 받을 의사가 없다면 MT는 '200 OK'가 아닌 다른 Failure message를 생성하여 상대방에게 보내야 한다.

4.3.6 RESPONDING

MT가 '200 OK'를 보내고 나서 MO로부터 'ACK'를 받아야 미디어 채널을 열 수 있다. '200 OK'를 보냈지만 네트워크나 서버의 error로 signal 데이터가 손실되어 MO가 받지 못하였을 경우 MT는 몇 차례 재전송한 후 timeout된다. 이때 CLEARING으로 전이한다.

4.3.7 ACTIVE

MO와 MT가 음성이나 영상과 같은 미디어 데이터를 주고받을 수 있는 상태이다. 또한, SIP signal중 'BYE' request를 이용하여 전화연결을 해제할 수 있는 상태이다.

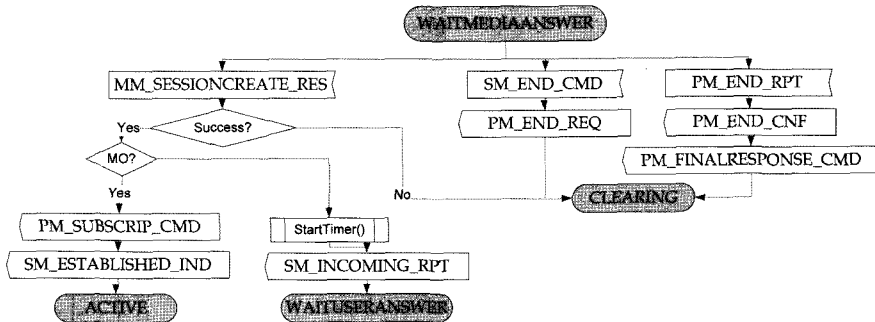


그림 8 WAITMEDIAANSWER state의 전이과정

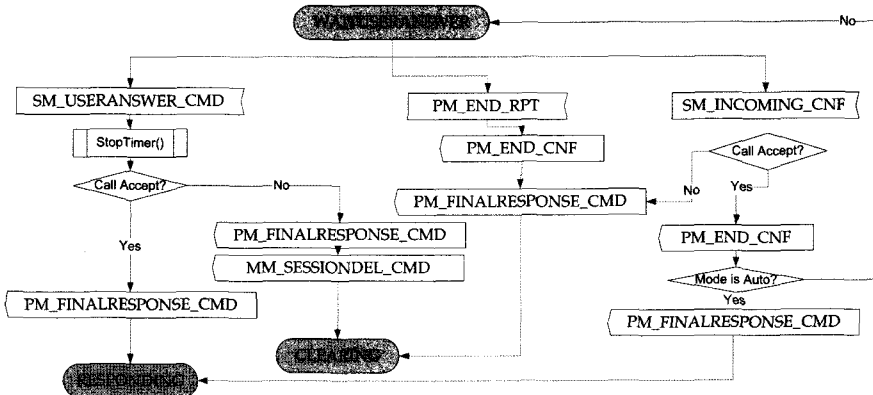


그림 9 WAITUSERANSWER state의 전이과정

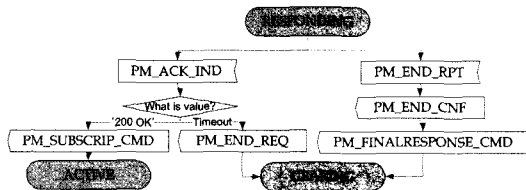


그림 10 RESPONDING state의 전이과정

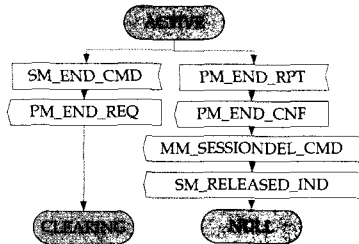


그림 11 ACTIVE state의 전이과정

4.4 Seven States를 이용한 전화연결 과정

MO와 MT가 Proxy Server에 REGISTER과정을 끝내고 전화연결의 유효한 상태가 되면 NULL로 초기화된다. MO가 원하는 상대를 선택하여 MT에 전화를 걸면 'INVITE' request를 보내고 ESTABLISHING State가 된다. MT는 이를 받고 요구되는 미디어 정보를 해석하여 MO와 미디어 채널 연결 가능 여부를 판단한다. 이때 MT는 WAITMEDIAANSWER state가 된다. MT가 해석할 수 없는 미디어 채널 혹은 미디어 코덱을 요구할 경우 MT는 Failure message를 보내야 하므로 CLEARING state로 간다. NULL상태로 바로 가지 않고 CLEARING state에서 SIP signal 특성상 주고받아야 하는 message를 걸러주어 시스템에 예외상황을 막는 역할을 한다. MT가 해석 가능한 미디어 채널을 요구할 경우 자신의 시스템 정보를 SDP message에 실어

서 response message를 보내고 MT는 RESPONDING state로 전이한다. 'INVITE'에 대해 '200 OK' response를 보냈더라도 ACK message를 받아야지 Call 연결이 성공하기 때문에 RESPONDING state에서 MO로부터의 ACK message를 기다린다. 한편 MT로부터 response를 받은 MO는 ACK message를 보낸 후 '200 OK'일 경우 미디어 채널을 생성 및 해제 관리하는 MM 모듈에 세션 생성 이벤트를 보내어 데이터를 받을 준비를 한다.

각 State는 FSM(Finite State Machine)으로 관리한다. FSM은 유한한 개수의 상태들을 가진 하나의 기계이고, 그 상태들 중 하나가 현재 상태인 것이다. FSM은 입력을 받고 어떠한 상태전이함수에 기반해서 현재 상태에서부터 출력상태로의 상태전이를 일으킨다. 그리고 출력상태는 새로운 현재 상태가 된다. FSM table을 두어 현재 상태에서 유효한 이벤트를 정의해 두어 해당 이벤트가 입력되면 조건에 맞는 상태로 전이한다. FSM에서는 이전의 상태를 읽어 올 수도 있고, Preprocessing Table을 두어 현재 상태에서 우선적으로 처리되어야 하는 이벤트를 실시간으로 받아 들여 시스템의 치명적인 오류를 막을 수 있다. 즉, User Termination 이거나 Network Close에 관련된 이벤트는 모든 상태에서 미리 정의되지 않았더라도 우선적으로 시스템에서 처리 및 안정적인 복구를 필요로 한다. 따라서 FSM의 Preprocessing Table에서 처리되도록 하는 것이 바람직하다.

5. 시스템 검증 및 구현

SIP 프로토콜을 이용하여 전화연결 후 응용서비스를 하기 위하여 시스템은 안정적으로 설계되어야 하며 어떠한 예외상황에서도 치명적인 오류뿐만 아니라 작은 오류를 일으켜서는 안 된다. 이를 검증하기 위하여 본

논문에서 제시하는 이벤트 기반의 Seven States구조는 WCDMA향 IP기반의 핸드폰 단말과 LG전자에서 자체 개발한 SIP Proxy서버를 인터넷에 물려서 실험하였다. SIP Proxy서버에 단말의 등록을 위한 Registration서버를 함께 돌아가도록 하였다. 단말에서 구현된 서비스로는 OMA(Open Mobile Appliance) Working Group으로 활동하고 있는 PoC(Push to Talk Over Cellular)를 구현하여 OMA-EICS-PoC-Client-V1\_0-20050729-D에서 제시하는 Test Items를 실험하여 표 2, 표 3과 같은 결과를 보인다[15]. OMA-EICS-PoC-Client-V1\_0-20050729-D는 OMA에서 제시하는 시스템 안정성 및

기능시험을 할 수 있는 자료로써 industry에서 자사 개발제품을 검증하는 KPI(Key Performance Indicator)로 많이 사용된다. 또한, 기존의 vovida.org에서 제시하는 session 설계방식을 이용하여 검증 가능한 OMA Item을 선별하여 결과를 보인다.

표 2는 Control Plan에서 검증되어야 하는 Item list이고 표 3은 User Plan관련 Item list이다. OMA-EICS-PoC-Client-V1\_0-20050729-D의 Control Plan은 PoC Client Originating Procedures(PoCCPSpec-COP), POC Service Settings procedure(PoCCPSpec-CSP), POC Session Initiation(PoCCPSpec-CSI), POC Ses

표 2 Client Enabler Implementation Statement (Control Plane)

OMA Item	OMA Function	OMA Ref.	Test Results	
			7 states	vovida
PoCCPSpec-COP-C-001	General registration to SIP/IP Core	6.1.1.1	Passed	Passed
PoCCPSpec-CSP-C-001	PoC Service Setting Procedure - initiation of SIP PUBLISH request	6.1.2	Passed	Passed
PoCCPSpec-CSP-C-002	Indication of setting results after receiving SIP 200OK response	6.1.2	Passed	Passed
PoCCPSpec-CSI-C-001	General Initiation - SIP INVITE request	6.1.3.1	Passed	Passed
PoCCPSpec-CSI-C-002	SIP Session Timer	6.1.3.1	Passed	N/A
PoCCPSpec-CSI-C-003	Pre-established Session establishment	6.1.3.2	Passed	N/A
PoCCPSpec-CSI-C-004	PoC Client Pre-established Session initiation	6.1.3.2.1	Passed	N/A
PoCCPSpec-CSI-C-005	PoC Client Pre-established Session initiation with example for Unconfirmed Indication for originating flow	6.1.3.2.1	Passed	N/A
PoCCPSpec-CSI-C-006	PoC Client Ad-hoc PoC Group Session and 1-1 PoC Session initiation (receiving SIP REFER request)	6.1.3.2.2	Partial	N/A
PoCCPSpec-CSI-C-007	Display of indication to PoC User of acceptance/rejection of PoC Session	6.1.3.2.2 6.1.3.2.3	Passed	Passed
PoCCPSpec-CSI-C-009	PoC Client releases a Pre-established Session	6.1.3.2.4	Passed	N/A
PoCCPSpec-CSI-C-010	Establishment of an On-Demand Session	6.1.3.3	Passed	N/A
PoCCPSpec-CSI-C-011~015	Ad-hoc PoC Group Session and 1-1 PoC Session setup (receiving request and response)	6.1.3.3.1	Partial	Partial
PoCCPSpec-CSI-C-016	Pre-arranged PoC Group Session setup	6.1.3.3.2	Partial	Partial
PoCCPSpec-CSI-C-017	Chat PoC Group Session setup	6.1.3.3.2	Partial	Partial
PoCCPSpec-CSM-C-001~003	General Procedures - SIP INVITE/UPDATE request	6.1.4.1	Passed	Passed
PoCCPSpec-CSM-C-004~005	PoC Client placing media on/off hold	6.1.4.2	Passed	Partial
PoCCPSpec-CSM-C-006	User plane adaptation: Modifying media capabilities during an ongoing PoC session	6.1.4.4	Passed	N/A
PoCCPSpec-CSM-C-007~008	SIP session refresh and Expiration	6.1.4.5	Passed	N/A
PoCCPSpec-CRS-C-001	On-demand established session	6.1.5.1	Passed	N/A
PoCCPSpec-CRS-C-002	Pre-established session	6.1.5.2	Passed	N/A
PoCCPSpec-CLS-C-001~002	Leaving a PoC session - on-demand case: Release the floor and BYE	6.1.6.1	Passed	Partial
PoCCPSpec-CLS-C-003	Leaving a PoC session - pre-established case: Generate SIP REFER/	6.1.6.2	Passed	N/A
PoCCPSpec-CLS-C-004	Leaving a PoC session - pre-established case: receive SIP NOTIFY	6.1.6.2	Passed	N/A
PoCCPSpec-CLS-C-005	Leaving a PoC session - display information contained in SIP NOTIFY	6.1.6.2	Passed	N/A
PoCCPSpec-CUO-C-001	PoC Client Adding a User to a PoC Session (SIP REFER)	6.1.7	Passed	N/A
PoCCPSpec-CUO-C-002	PoC Client Sending an Instant Personal Alert (sending SIP MESSAGE)	6.1.8	Passed	Partial



PoCCPSpec-CUO-C-003	PoC Client Sending a Group Advertisement (sending SIP MESSAGE)	6.1.9	Partial	N/A
PoCCPSpec-CUO-C-004	PoC Client Subscription to the Conference State Event Package (sending SIP SUBSCRIBE request) for Pre-established Session	6.1.10	Passed	Passed
PoCCPSpec-CUO-C-005	PoC Client Subscription to the Conference State Event Package (sending SIP SUBSCRIBE request) for On-Demand Session	6.1.10	Passed	Passed
PoCCPSpec-CUO-C-006	Canceling a PoC session initiation - on-demand case	6.1.11.1	Passed	Passed
PoCCPSpec-CUO-C-007	Simultaneous Session Control Procedures	6.1.12	Passed	Passed
PoCCPSpec-CUO-C-008	Simultaneous Session control: PoC Client Setting Session Priority indicating with SDP attribute	6.1.12.1	Passed	Passed
PoCCPSpec-CUO-C-009	Simultaneous Session control: PoC Client handling for PoC session locking in a particular PoC Session	6.1.12.2	Passed	Passed
PoCCPSpec-CTP-C-001~005	PoC Client invited to a PoC Session: General INVITE procedure, Automatic Answer Mode Or Manual Answer Mode, Auto-answer, Manual-answer, Manual-answer-override	6.2.1.1~ 6.2.1.4	Passed	Partial
PoCCPSpec-CTP-C-006	PoC Client receiving a PoC Session Modification Request (send 200OK after receiving UPDATE or re-INVITE)	6.2.2	Passed	Passed
PoCCPSpec-CTP-C-007~008	PoC Client receiving a PoC Session Termination Request: On-demand case and Pre-established case	6.2.3.1	Passed	Passed
PoCCPSpec-CTP-C-009~010	PoC Client receiving an Instant Personal Alert: receiving SIP MESSAGE, initiating 1-1 PoC session	6.2.4	Passed	Passed
PoCCPSpec-CTP-C-011	PoC Client Receiving a Group Advertisement	6.2.5	Partial	N/A
PoCCPSpec-CTP-C-012	PoC Session Cancel request: On-demand case (receiving SIP CANCEL)	6.2.6.1	Passed	Passed
PoCCPSpec-CTP-C-013	Simultaneous Session Control Procedures	6.2.7	Passed	Passed
PoCCPSpec-CTP-C-014	PoC Client setting the PoC Session priority in SDP payload	6.2.7.1	Passed	Passed
PoCCPSpec-CTP-C-015	PoC Client handling for PoC session locking in a particular PoC Session	6.2.7.2	Partial	Partial

### 표 3 Client Enabler Implementation Statement (User Plane)

OMA Item	OMA Function	OMA Ref.	Test Results	
			7 states	voida
PoC_UserPlaneV1-UTR-C-002	Support of UDP according to rules and procedures as specified in [RFC768].	5.2	Passed	Partial
PoC_UserPlaneV1-UTR-C-003	Handling of Port Numbers.	5.2.1	Passed	Passed
PoC_UserPlaneV1-UTR-C-004	Support of RTP according to rules and procedures as specified in [RFC3550].	5.3	Passed	Passed
PoC_UserPlaneV1-UTR-C-005	Support of RTCP according to the rules and procedures specified in [RFC3550] at the minimum the reception of RTCP packets.	5.4	Passed	Passed
PoC_UserPlaneV1-UTR-C-006	RTCP packets only contain the mandatory parts of RTCP (according to and procedures as specified in [RFC3550]), which are required for that specific RTCP compound packet.	5.4	Passed	Passed
PoC_UserPlaneV1-UTR-C-007	Support of sending of TBCP messages to the same UDP port as the other RTCP packets.	5.4	Passed	Passed
PoC_UserPlaneV1-UTR-C-008	Does not send TBCP messages as compound packets.	5.4	Passed	Passed
PoC_UserPlaneV1-UTR-C-009	Supports the creation, modification and/or processing of the content in RTCP packets.	5.4	Passed	Partial
PoC_UserPlaneV1-UTR-C-010	To reduce potential degradation of the quality of the media transmission, the PoC Client does not schedule transmission of RTCP packets or TBCP messages during a Talk Burst.	5.4	Passed	N/A
PoC_UserPlaneV1-UTR-C-011	Supports only mandatory parts of RTPC according to the rules and procedures specified in [RFC3550]	5.4	Passed	N/A
PoC_UserPlaneV1-UTR-C-012	Does not send RTCP BYE packets when the PoC Session is released.	5.4	Passed	Passed
PoC_UserPlaneV1-UTB-C-001	Supports basic talk burst control	6.2.5	Passed	Passed

PoC_UserPlaneV1-UMC-C-002	Supports the transmission of RTCP SR/RR compound packets.	7.1.2	Passed	Passed
PoC_UserPlaneV1-UMC-C-005	Support of Media parameter negotiation.	7.2	Passed	Partial
PoC_UserPlaneV1-UMC-C-006	Support of User Plane adaptation.	7.2	Passed	N/A
PoC_UserPlaneV1-UMC-C-007	Changes the voice frame packetization or voice codec mode by Out-band signaling using SDP payload within SIP messages.	7.2	Passed	N/A
PoC_UserPlaneV1-UMC-C-008	Changes the voice frame packetization or voice codec mode by In-band signaling using Codec Mode Request (CMR) field of AMR payload.	7.2	Passed	N/A
PoC_UserPlaneV1-UMC-C-009	Initiate User Plane adaptation (in-band or out-band) triggered by e.g roaming to the system with different media capabilities.	7.3.1	Passed	N/A
PoC_UserPlaneV1-UMC-C-010	Supports Media on hold/off hold	7.4.1	Passed	Partial
PoC_UserPlaneV1-UMC-C-013	Supports the sending of RTP Media packets to any of the Simultaneous PoC Sessions according to the PoC User selection.	7.5.1	Passed	Passed
PoC_UserPlaneV1-UMC-C-014	Supports RTP Media Session release of the Pre-established Session.	7.5.1	Passed	Partial
PoC_UserPlaneV1-UMC-C-015	Support of Media transfer	7.7.1	Passed	Passed
PoC_UserPlaneV1-UMC-C-016	Supports media buffering for handling of variable latency in incoming RTP Media packets.	7.8.1	Passed	N/A
PoC_UserPlaneV1-UMC-C-017	Support coding and decoding of Media	7.9	Passed	Passed

sion Modification(PoCCPSpec-CSM), PoC Client Re-joining a PoC Session(PoCCPSpec-CRS), PoC Client Leaving a PoC Session(PoCCPSpec-CLS), PoC Session unrelated operations for PoC Client Initiating Procedures(PoCCPSpec-CUO), PoC Client Terminating Procedures(PoCCPSpec-CTP)로 구성한다. User Plan은 Transport(PoC\_UserPlaneV1-UTR), Talk Burst Control(PoC\_UserPlaneV1-UTB), Media control (PoC\_UserPlaneV1-UMC), Talker identification(PoC\_UserPlaneV1-UID), Timers(PoC\_UserPlaneV1-UTI)으로 구성한다. 표 2와 3에서 Test Results는 OMA PoC 표준스펙에 맞추어 구현 후 실험한 결과를 보인다. 본 논문에서의 설계방식과 vovida.org에서 제시하는 방식의 결과를 비교한다.

## 6. 결론 및 향후 연구 과제

최근에는 핸드폰과 같은 소형 단말에서만 아니라 Digital TV나 정보가전에서도 SIP를 이용한 다양한 응용시나리오를 적용하고 있다. 본 논문은 단말에서 응용서비스를 효율적으로 개발하기 위하여 Seven States를 설정하고 발생할 수 있는 이벤트를 정의하여 SIP프로토콜로 Call-Setup을 하는 모든 응용시나리오에 적용할 수 있다. 응용서비스에 따라 다양한 예외상황이 발생하고 단말의 치명적인 오류에 대해 안정적으로 처리할 수 있는 메커니즘을 보인다. PoC라는 응용서비스를 구현하여 OMA에서 제시하는 EICS의 모든 Item들을 pass하였다. OMA에서 제시하는 Test case가 PoC에 의존적이라는 한계는 있었지만 Seven States구조로 설계된 SIP 인터넷 전화연결 시스템은 효율적이고 확장

가능한 signal처리를 할 수 있다. 표 2, 3의 실험 결과와 같이 SIP 전화연결 시 다양한 예외상황을 볼 수 있는데 각각의 경우에서도 상대방과의 세션을 유지하면서 안정적인 미디어 전송이 가능하였다. 또한, 새로운 응용서비스가 추가되면 시나리오에 따른 이벤트를 추가로 정의하여 FSM에 적용하여 개발자가 쉽게 확장 가능한 구조이다.

향후 연구과제로는 새로운 응용서비스에 따른 추가 이벤트를 정형화 할 수 있는 구조설계를 하여 이벤트 추가 없이 User Interface API만 변경하는 구조를 설계할 것이다.

## 참고 문헌

- [1] J. Rosenberg, H.Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M/ Handley, E/ Schooler, "Session Initiation Protocol," IETF, RFC 3261, June 2002.
- [2] 허미영, 한재천, 현욱, 박선욱, 강신각, "SIP 기반 인터넷전화 서비스를 위한 사용자 에이전트의 설계 및 구현", 한국정보과학회논문지, 정보통신, 제32권 제3호, 2005년 6월.
- [3] Telecommunication Standardization Sector of ITU, "ITU-T Recommendation H.323-Packet based multimedia communications systems," Orlando, FL, Feb. 1998.
- [4] J. Rosenberg, "The Session Initiation Protocol (SIP) UPDATE Method," IETF, RFC 3311, Sep. 2002.
- [5] W. Marshall, "Private Session Initiation Protocol (SIP) Extensions for Media Authorization," IETF, RFC 3313, Jan. 2003.
- [6] Arkko, J. Torvinen, V. Camarillo, G., Niemi A. and Haukka T., "Security Mechanism Agreement for

- the Session Initiation Protocol," IETF, RFC 3329, 2003.
- [ 7 ] C. Huang, C. Lee and C. Zheng, "A Novel SIP-Based Route Optimization for Network Mobility," *IEEE Commun.*, Vol.24, No.9, pp. 1682-1691, Sept. 2006.
- [ 8 ] Salsano, S. Veltri, L. and Papalilo D., "SIP Security Issues: The SIP authentication procedure and its processing load," *IEEE Network*. Vol.16, No.6, pp. 38-44, Nov.-Dec. 2002.
- [ 9 ] A. Johnston and O. Levin, "Session initiation protocol call control Conferencing for User agents," IETF DRAFT: draft-ietf-sipping-cc-conferencing-06, 2004.
- [10] T. Eysers and H. Schulzrinne, "Predicting internet telephony call setup delay," in *Proc. IP Telephony Workshop*, Berlin, Germany, Apr. 2000.
- [11] vovida software, <http://www.vovida.org/protocols/downloads/sip/>
- [12] R. V. Prasad, R. Hurni, and H. S. Jamadagni, "A scalable distributed VoIP conferencing using SIP," in *Proc. 8th IEEE Int. Symp. Comput. Commun.*, Jun./Jul. 2003, pp. 608-613.
- [13] B. Campbell, R. Mahy and C. Jennings, "MSRP: The Message Session Relay Protocol," IETF DRAFT: draft-ietf-simple-message-sessions-10, Feb. 2005.
- [14] M. Handley and V. Jacobson, "SDP: Session Description Protocol," IETF, RFC 2327, April 1998.
- [15] Open Mobile Alliance, <http://www.openmobilealliance.org/>



신 용 경

1999년 12월 경북대학교 컴퓨터과학(학사). 2001년 2월 경북대학교 컴퓨터과학(석사). 2002년 12월 경북대학교 컴퓨터과학(박사수료). 2003년 2월~현재 LG전자기술원 선임연구원. 관심분야는 MPEG-4 콘텐츠, 이동 멀티미디어, MPEG-4 실시간 스트리밍, 멀티미디어 콘텐츠 저작 등



김 상 욱

1979년 경북대학교 컴퓨터공학(학사). 1981년 서울대학교 컴퓨터과학(석사). 1989년 서울대학교 컴퓨터과학(박사). 현재 경북대학교 전자전기컴퓨터학부 교수. 관심분야는 모바일 멀티미디어 시스템, 멀티미디어 콘텐츠 저작 및 인간과 컴퓨터의 상호작용