

SIP 기반 사용자 에이전트 설계 및 구현

한재천* · 강신각
한국전자통신연구원

Design and Implement of SIP-based User Agent

Jae-cheon Han* · Sin-gak Kang
Electronics and Telecommunications Research Institute
E-mail : { jupiter* , sgkang } @etri.re.kr

요 약

SIP(Session Initiation Protocol) 프로토콜은 구조가 간단하고 다양한 애플리케이션을 개발할 수 있는 등의 장점을 가지고 있기 때문에 차세대 네트워크에서 호 설정을 위한 사실상의 표준으로 자리 잡아가고 있다. SIP는 계층적 구조를 갖는 프로토콜로서, SIP의 일부 계층은 대부분의 SIP 응용 애플리케이션에서 공통적으로 사용될 수 있다. 본 논문에서는 SIP 애플리케이션 개발에 효과적으로 사용될 수 있는 SIP 공통모듈에 대한 간략한 설명과 이를 이용한 사용자 에이전트의 설계 및 구현에 대하여 설명한다.

키워드

RFC3261, SIP, VoIP, UA, UAC, UAS, User Agent

I. 서 론

SIP 프로토콜은 세션 설정을 필요로 하는 다양한 인터넷 응용 프로그램을 위한 프레임워크를 제공하는 프로토콜로서 구조가 간단하고 다양한 애플리케이션을 용이하게 결합시킬 수 있는 장점을 갖고 있기 때문에 차세대 네트워크(NGN: Next Generation Network)에서 멀티미디어, 음성데이터 통합(VoIP)등을 구현하기 위한 사실상의 표준으로 자리를 굳히고 있다.

본 논문에서는 SIP 프로토콜 계층 중 트랜잭션 계층, 전송 계층, 문법 및 인코딩 계층의 처리를 수행해 주는 SIP 공통 모듈을 사용하여 기본적인 호 설정 및 해제, 호 재설정이 가능하도록 설계·구현한 인터넷 텔레포니를 위한 사용자 에이전트에 대하여 기술하고자 한다. 2장에서는 SIP 프로토콜과 SIP 공통 모듈에 대해 설명하고, 3장에서는 인터넷 텔레포니를 위한 사용자 에이전트의 외 구조에 대하여 상세히 설명한다. 결론 및 향후 연구 과제는 4장에서 기술한다.

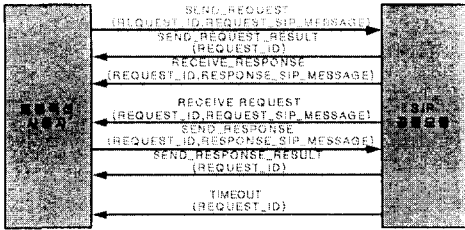
II. SIP 프로토콜과 공통모듈

SIP 프로토콜은 인터넷 전화와 같은 멀티미디어 세션을 설정, 변경 그리고 종료시킬 수 있는 응용 계층의 제어 프로토콜이며, Transaction User, Transaction Layer, Transport Layer, Syntax & Encoding Layer의 4계층으로 구성된다. <표1>은 SIP 프로토콜 계층의 기능을 요약한 것이다.

계 층	기 능
문법 및 인코딩 (Syntax & Encoding)	- SIP 메시지 문법 정의 - 메시지 인코딩 - 메시지의 오류 검사
전송 (Transport)	- 네트워크를 통한 메시지 송신 및 수신
트랜잭션 (Transaction)	- 메시지 재전송 - 응답/요청 메시지 매칭 - 메시지 타임아웃
트랜잭션 사용자 (Transaction User)	- 메시지 목적지 위치 및 전송 프로토콜 결정 - 트랜잭션 사용자 기능에 따른 동작

<표4> SIP 프로토콜 계층의 기능 요약

SIP 공통모듈은 Stateless Proxy를 제외한 거의 모든 SIP 응용 애플리케이션에서 공통적으로 사용될 수 있는 문법 및 인코딩 계층, 전송 계층, 트랜잭션 계층의 기능과 트랜잭션 사용자의 일부 기능을 묶은 모듈로서 <그림 1>과 같은 인터페이스를 제공한다.



<그림 1> SIP 공통모듈의 인터페이스

SIP 공통모듈이 제공하는 인터페이스는 다음과 같다.

■ SEND_REQUEST

트랜잭션 사용자가 요청 메시지를 전송할 때 사용하는 인터페이스 함수이다.

② SEND_RESPONSE

트랜잭션 사용자가 응답 메시지를 전송할 때 사용하는 인터페이스 함수이다.

③ SEND_REQUEST_RESULT

SIP 공통모듈에서 요청 메시지를 전송하고 그 결과를 트랜잭션 사용자에게 통보하기 위해 호출하는 인터페이스 함수이다.

④ SEND_RESPONSE_RESULT

SIP 공통모듈이 응답 메시지를 전송하고 그 결과를 트랜잭션 사용자에게 통보하기 위해 호출하는 인터페이스 함수이다.

⑤ RECEIVE_REQUEST

SIP 공통모듈에서 요청 메시지를 수신하였을 때 이를 트랜잭션 사용자에게 통보하기 위하여 호출하는 인터페이스 함수이다.

⑥ RECEIVE_RESPONSE

SIP 공통모듈에서 응답 메시지를 수신하였을 때 이를 트랜잭션 사용자에게 통보하기 위하여 호출하는 인터페이스 함수이다.

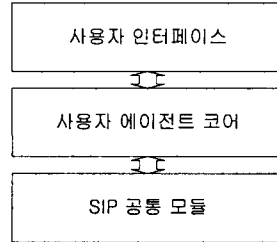
⑦ TIMEOUT

SIP 공통모듈에서 트랜잭션 타임아웃이 발생했을 때 트랜잭션 사용자에게 통보하기 위하여 호출하는 인터페이스 함수이다.

III. 인터넷 텔레포니를 위한 사용자 에이전트

본 논문에서 설계한 사용자 에이전트는 SIP 공통 모듈, 사용자 에이전트 코어 모듈, 그리고 사용자 인터페이스 모듈로 구성되어 있다. 사용자 인터페이스 모듈은 사용자와 상호작용을 담당하는 모

듈로서 사용자 인터페이스 부분만을 변경하여 Residential Gateway나 IP Phone과 같은 다양한 장치에 사용할 수 있도록 하기 위하여 하나의 독립된 모듈로 분리하였다. <그림2>는 사용자 에이전트의 구성을 간략히 보여준다.



<그림 2> 사용자 에이전트의 구조

✓ 사용자 에이전트 코어와 사용자 인터페이스 모듈 사이의 인터페이스

사용자 에이전트 코어 모듈에서 사용자 인터페이스 모듈과 인터페이스하기 위하여 다음과 같은 이벤트 메시지와 콜백 함수를 정의하였다.

■ 이벤트 메시지

- ① CallSetup
호 설정 요청 이벤트 메시지
- ② CallEnd
호 해제 요청 이벤트 메시지
- ③ UserRsp
수신된 호에 대한 사용자의 수락/거부를 나타내는 이벤트 메시지
- ④ CallAuthentication
호에 대한 인증 시도 요청 이벤트 메시지
- ⑤ CancelCallAuthentication
호에 대한 인증 취소 요청 이벤트 메시지

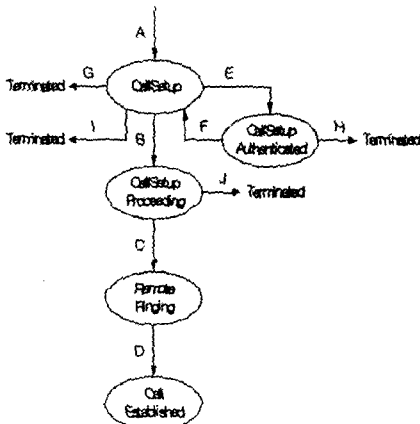
■ 콜백 함수

- ① UI_CallTrying
INVITE를 보내고 100 응답을 받았을 때 호출한다.
- ② UI_CallRemoteRinging
INVITE를 보내고 180 응답을 받았을 때 호출한다.
- ③ UI_CallRinging
INVITE를 받았을 때 호출한다.
- ④ UI_CallAccept
사용자가 걸려온 호를 수락했을 때 200 응답을 보내고 호출한다.
- ⑤ UI_CallReject
사용자가 걸려온 호를 거절했을 때 603 응답을 전송하고 호출한다.
- ⑥ UI_CallConnected
200 응답을 전송하고 ACK를 수신했을 경우 또는 INVITE를 전송하고 200 응답을 받

- 왔을 경우에 호출한다.
- ⑦ UI_CallDisconnecting
사용자의 호 해제 요청이 있을 경우에 BYE를 전송하고 호출한다.
 - ⑧ UI_CallDisconnected
BYE를 전송하고 200 응답을 받았을 경우 또는 BYE를 수신했을 경우에 호출한다.
 - ⑨ UI_CallCanceled
CANCEL을 수신했을 때 호출한다.
 - ⑩ UI_CallFailure
메시지 전송 실패 등으로 인해 호 연결을 더 이상 진행할 수 없을 때 호출한다.
 - ⑪ UI_CallAuthenticate
INVITE 메시지를 전송하고 인증 요구를 받았을 때 호출한다.

✓ 사용자 에이전트 코어 모듈의 상태도

사용자 에이전트 코어 모듈은 호 설정 및 해제, 호 재설정을 위하여 <그림3>~<그림6>과 같은 상태도를 사용한다.

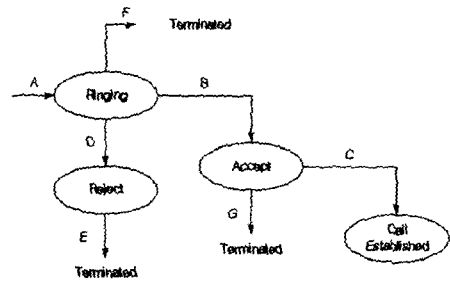


<그림 3> 발신자 호 설정 상태도

<그림3>는 발신자로서 호 설정을 하기 위한 상태도이며, 상태도의 stimulus/action은 다음과 같다.

- ① A: CallSetup
SEND_REQUEST(INVITE) 호출
- ② B: RECEIVE_RESPONSE(INVITE,100)
UI_CallTrying 호출
- ③ C: RECEIVE_RESPONSE(INVITE,180)
UI_CallRemoteRinging 호출
- ④ D: RECEIVE_RESPONSE(INVITE,200)
UI_CallConnected 호출
- ⑤ E: RECEIVE_RESPONSE(INVITE,407)
UI_CallAuthenticate 호출
- ⑥ F: CallAuthentication

- SEND_REQUEST(INVITE) 호출
- ⑦ G: SEND_REQUEST_RESULT(Fail)
UI_CallFailure 호출
 - ⑧ H: CancelCallAuthentication
UI_CallFailure 호출
 - ⑨ I: RECEIVE_RESPONSE(INVITE, Non 2xx, except for 40/407)
UI_CallFailure 호출



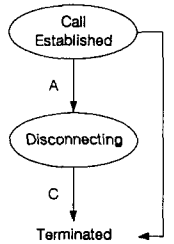
<그림 4> 수신자 호 설정 상태도

<그림4>는 수신자로서 호 설정을 하기 위한 상태도이며, 상태도의 stimulus/action은 다음과 같다.

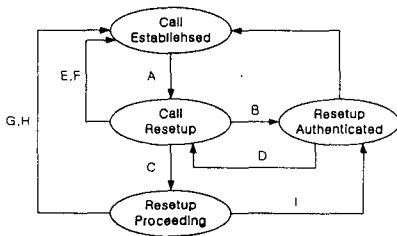
- ① A: RECEIVE_REQUEST(INVITE)
UI_CallRinging 호출
- ② B: UserRsp(Accept)
SEND_RESPONSE(INVITE,180)
- ③ C: RECEIVE_REQUEST(ACK)
UI_CallConnected 호출
- ④ D: UserRsp(Decline)
UI_CallReject 호출
- ⑤ E: RECEIVE_REQUEST(ACK)
- ⑥ F: SEND_RESPONSE_RESULT(Fail)
UI_CallFailure 호출
- ⑦ G: SEND_REQUEST_RESULT(Fail)
UI_CallFailure 호출

<그림5>는 호를 종료하기 위한 상태도이며, 상태도의 stimulus/action은 다음과 같다.

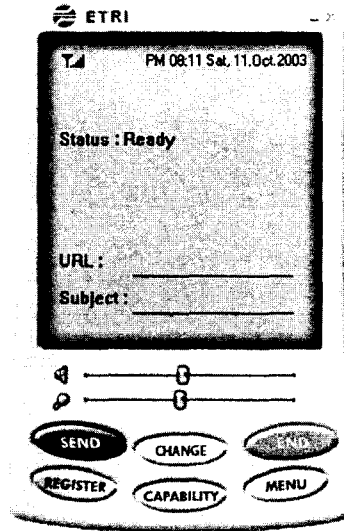
- ① A: CallEnd
SEND_REQUEST(BYE) 호출
UI_CallDisconnecting 호출
- ② B: RECEIVE_REQUEST(BYE)
UI_CallDisconnected 호출
- ③ C: RECEIVE_RESPONSE(BYE,ANY)
UI_CallDisconnected 호출



<그림 5> 호 종료 상태도



<그림 6> 호 재설정 상태도



<그림 7> 사용자 에이전트 실행화면

<그림6>은 호를 재설정을 하기 위한 상태도이며, 상태도의 stimulus/action은 다음과 같다.

- ① A: CallResetup
SEND_REQUEST(INVITE) 호출
- ② B: RECEIVE_RESPONSE(INVITE,407)
UI_CallAuthenticate 호출
- ③ C: RECEIVE_RESPONSE(INVITE,1xx)
-
- ④ D: CallAuthentication
SEND_REQUEST(INVITE) 호출
- ⑤ E: RECEIVE_RESPONSE(INVITE,200)
-
- ⑥ F: RECEIVE_RESPONSE(INVITE, Non 2xx,
except for 40/407)
-
- ⑦ G: RECEIVE_RESPONSE(INVITE,200)
-
- ⑧ H: RECEIVE_RESPONSE(INVITE, Non
2xx, except for 40/407)
-
- ⑨ I: RECEIVE_RESPONSE(INVITE,407)
UI_CallAuthenticate 호출

✓ 사용자 에이전트 구현물

사용자 에이전트 코어 모듈은 ANSI C 기반으로 작성되었으며 MS Visual C++ 6 컴파일러를 사용하여 컴파일하였다. 사용자 인터페이스 모듈은 MS Windows 2000 운영체제에서 GUI로 구현하였다. <그림 7>은 사용자 에이전트의 실행 화면을 보여준다.

IV. 결론

본 논문에서는 SIP 공통 모듈을 사용하여 설계·구현한 SIP 기반 인터넷 텔레포니를 위한 사용자 에이전트에 대하여 기술하였다.

본 논문에서 구현한 사용자 에이전트는 호 설정 및 해제와 호 재설정 기능만이 가능하도록 되어있기 때문에 향후 REGISTER, OPTIONS 등 부가적인 기능을 추가 개발할 예정이다.

참고 문헌

- [1] IETF SIP Working Group, <http://www.ietf.org/html.charters/sip-charter.html>
- [2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. "SIP: session initiation protocol" IETF RFC 3261, June 2002
- [3] J. Rosenberg and H. Schulzrinne, "Guidelines for Authors of Extensions to the Session Initiation Protocol (SIP)" IETF draft-ietf-sip-guidelines-06, Nov. 4, 2002