

SIP 공통 모듈의 설계 및 구현

한재천⁰ · 강신각
한국전자통신연구원
{jupiter⁰, sgkang}@etri.re.kr

Design and Implement of SIP Common Module

JaeCheon Han⁰ SinGak Kang
Electronics and Telecommunications Research Institute

요 약

VoIP(Voice over IP) 기술들 중에서 IETF(Internet Engineering Task Force)에서 제안한 텍스트 기반의 SIP(Session Initiation Protocol) 프로토콜은 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 있도록 설계 및 구현된 SIP 공통모듈에 대하여 설명한다.

1. 서 론

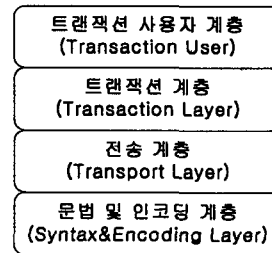
인터넷의 사용이 급속히 확산되면서 “음성과 데이터의 통합”이라 불리는 VoIP와 관련된 기술들에 대한 연구가 활발히 진행되고 있다. VoIP 기술은 ITU-T의 H.323 프로토콜과 IETF의 SIP 프로토콜을 중심으로 표준화가 수행되고 있다[1]. 현재 인터넷 전화 서비스를 위하여 H.323 프로토콜을 기반으로 하는 장비와 단말기가 많이 보급되어 있는 상태이지만 SIP 프로토콜은 H.323에 비하여 구조가 간단하고 다양한 애플리케이션을 용이하게 결합시킬 수 있는 장점을 갖고 있기 때문에 차세대 네트워크(NGN: Next Generation Network)에서 멀티미디어, 음성데이터통합(VoIP) 환경을 구현하기 위한 사실상의 표준으로 자리를 굳히고 있다.

SIP 프로토콜은 트랜잭션 사용자(Transaction User), 트랜잭션 계층(Transaction Layer), 전송 계층(Transport Layer), 문법 및 인코딩 계층(Syntax & Encoding Layer)의 4 계층으로 설명될 수 있는 프로토콜이다. SIP 프로토콜 계층 중에서 트랜잭션 사용자를 제외한 다른 모든 계층은 대부분의 SIP 응용 애플리케이션에서 공통적으로 사용된다[2]. 공통된 계층을 재사용이 가능하도록 모듈화 한다면 SIP 응용 애플리케이션을 보다 쉽고 빠르게 개발할 수 있을 것이다.

본 논문에서는 SIP 프로토콜이 갖고 있는 계층적 구조를 고려해 대부분의 SIP 애플리케이션 개발에서 공통적으로 사용할 수 있는 SIP 모듈에 관하여 기술하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 계층 중심으로 SIP 프로토콜에 관해 간략히 기술하고, 3장에서는 SIP 프로토콜의 계층 구조를 고려한 SIP 공통모듈의 설계 및 구현에 관하여 기술한다. 결론 및 향후 연구 과제는 4장에서 기술한다.

2. SIP 프로토콜의 계층구조

SIP 프로토콜은 Transaction User, Transaction Layer, Transport Layer, Syntax & Encoding Layer의 계층으로 설명될 수 있다. <그림 1>은 SIP 프로토콜의 기능적 계층 구조를 보여준다.



<그림 1> SIP 프로토콜 계층

▶ 문법 및 인코딩 계층(Syntax & Encoding Layer)

문법 및 인코딩 계층은 SIP 프로토콜의 가장 마지막 계층으로 ABNF(Augmented Backus-Naur Form)을 사용하여 SIP 메시지의 문법을 정의하고 있으며, 여기서 정의된 문법을 이용하여 메시지의 오류 검사 등을 처리한다.

▶ 전송 계층(Transport Layer)

전송 계층에서는 네트워크를 통해서 클라이언트에서 어떻게 요청 메시지를 보내고 응답 메시지를 받을 것인가, 서버에서 어떻게 요청 메시지를 받고 응답 메시지를 보낼 것인가 등에 대해 담당한다.

▶ 트랜잭션 계층(Transaction Layer)

트랜잭션 계층은 SIP 프로토콜의 근간을 이루는 트랜잭션을 다루는 계층으로 메시지의 재전송, 응답 메시지와 요청 메시지의 매칭, 그리고 메시지 타임아웃 등의 처리를 담당한다.

▶ 트랜잭션 사용자(TU: Transaction User)

트랜잭션 사용자는 요청 메시지를 보내고자 할 때 클라이언트 트랜잭션을 생성하여 목적지 IP, 포트, 전송 프로토콜과 함께 트랜잭션 계층에 전달한다. 요청 메시지를 받았을 때에는 SIP 엔티티(Entity)의 기능과 요청 메시지의 메소드에 따라 적절한 동작을 취하고, 응답 메시지를 생성해서 전송한다. 스테이트리스 프락시(Stateless Proxy)를 제외한 유저 에이전트 클라이언트(UAC), 유저 에이전트 서버(UAS), 프락시 서버 등은 트랜잭션 사용자에게 해당한다.

3. SIP 공통모듈

3.1 SIP 공통모듈과 트랜잭션 사용자 사이의 인터페이스

트랜잭션 사용자 계층은 트랜잭션 계층과 연결되어 있기 때문에 SIP 공통모듈과 트랜잭션 사용자 사이의 인터페이스는 트랜잭션 계층과 트랜잭션 사용자 계층 사이의 인터페이스와 동일하게 정의할 수 있다.

트랜잭션 계층에서는 <표 1>과 같은 사건이 발생하면 이를 트랜잭션 사용자 계층에 통보하도록 되어 있다.

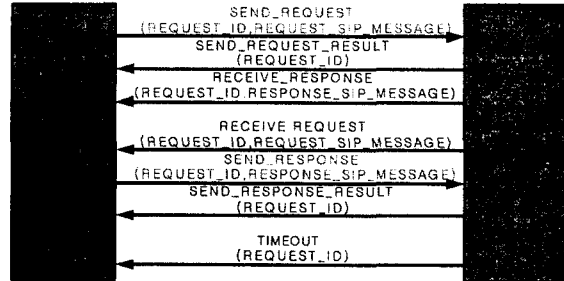
트랜잭션 종류	사건
Invite 클라이언트 트랜잭션	<ul style="list-style-type: none"> ✓ 응답 메시지 수신 ✓ 전송 에러 발생 ✓ 타이머 B 만료
Invite 서버 트랜잭션	<ul style="list-style-type: none"> ✓ Invite 메시지 수신 ✓ 전송 에러 발생 ✓ 타이머 H 만료
Non-Invite 클라이언트 트랜잭션	<ul style="list-style-type: none"> ✓ 응답 메시지 수신 ✓ 전송 에러 발생 ✓ 타이머 F 만료
Non-Invite 서버 트랜잭션	<ul style="list-style-type: none"> ✓ Non-Invite 메시지 수신 ✓ 전송 에러 발생

<표 1> 트랜잭션 계층에서 발생하는 사건

RFC3263 "Locating SIP Servers"에서는 요청 메시지를 전송할 때 메시지가 전달될 목적지의 IP 주소, 포트번호, 전송 프로토콜 등을 결정하는 절차를 기술하고

있다[3]. 이는 본질적으로 모든 SIP 애플리케이션에서 필요한 기능이기 때문에 SIP 공통모듈에 포함시켰다.

트랜잭션 사용자 계층과 트랜잭션 계층 사이의 인터페이스는 <그림 2>와 같이 정의하였다.



<그림 2> 트랜잭션 사용자 계층과 SIP 공통모듈 사이의 인터페이스

SIP 공통모듈은 트랜잭션 사용자가 요청 메시지와 응답 메시지를 전송할 수 있는 인터페이스 함수를 제공하며, 요청 메시지마다 유일한 REQUEST_ID를 할당하여 트랜잭션 단위의 구분을 가능하도록 해준다.

트랜잭션 사용자는 SIP 공통 모듈로부터 메시지 전송 결과를 통보받을 수 있는 인터페이스 함수와 SIP 공통모듈이 SIP 메시지를 수신하였을 때 이를 통보받을 인터페이스 함수를 콜백 함수로 제공하여야 한다.

▶ SEND_REQUEST

트랜잭션 사용자가 요청 메시지를 전송할 때 사용하는 인터페이스 함수이다. 이 함수가 호출되면 SIP 공통모듈은 요청 메시지에 REQUEST_ID를 부여하고, 메시지 전송을 준비한다.

▶ SEND_REQUEST_RESULT

SIP 공통모듈에서 요청 메시지를 전송하고 그 결과를 트랜잭션 사용자에게 통보하기 위해 호출하는 인터페이스 함수이다. SEND_REQUEST 인터페이스에서 제공한 REQUEST_ID가 동시에 제공된다.

▶ RECEIVE_RESPONSE

SIP 공통모듈에서 응답 메시지를 수신하였을 때 이를 트랜잭션 사용자에게 통보하기 위하여 호출하는 인터페이스 함수이다. SEND_REQUEST 인터페이스 함수에서 할당한 REQUEST_ID를 동시에 제공한다.

▶ RECEIVE_REQUEST

SIP 공통모듈에서 요청 메시지를 수신하였을 때 이를 트랜잭션 사용자에게 통보하기 위하여 호출하는 인터페이스 함수이다. 수신한 메시지에 REQUEST_ID를 새로 부여하여 요청 메시지와 함께 제공한다.

▶ SEND_RESPONSE

트랜잭션 사용자가 응답 메시지를 전송할 때 사용하는 인터페이스 함수이다. SIP 공통모듈이 RECEIVE_REQUEST 인터페이스 함수를 통해 제공한 REQUEST_ID를 동시에 인자로 넘겨주어야 한다.

▶ SEND_RESPONSE_RESULT

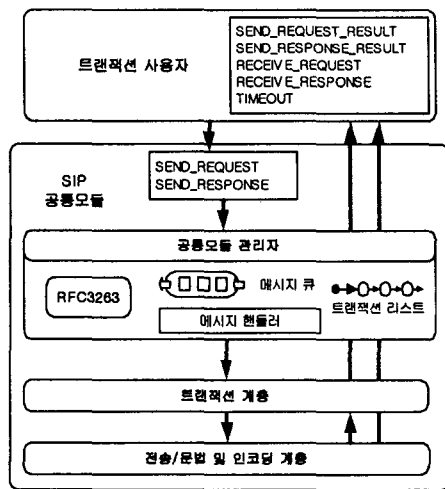
SIP 공통모듈이 응답 메시지를 전송하고 그 결과를 트랜잭션 사용자에게 통보하기 위해 호출하는 인터페이스 함수이다.

▶ TIMEOUT

SIP 공통모듈에서 트랜잭션 타임아웃이 발생했을 때 트랜잭션 사용자에게 통보하기 위하여 호출하는 인터페이스 함수이다. 타임아웃이 발생한 요청 메시지의 REQUEST_ID가 같이 전달된다.

3.2 SIP 공통모듈의 구조

SIP 공통모듈은 인터페이스 함수, 공통모듈 관리자, 트랜잭션 계층, 전송/문법 및 인코딩 계층으로 설계되었다. <그림 3>은 SIP 공통모듈의 구조를 보여준다.



<그림 3> SIP 공통모듈 구조

SIP 공통모듈을 구성하고 있는 각각의 구성요소들의 역할은 다음과 같다.

▶ 공통모듈 관리자

- SIP 공통모듈의 전반적인 제어 흐름 관리
- 트랜잭션의 관리(생성, 삭제, 매칭)

- 목적지 주소 계산(RFC3263)

▶ 트랜잭션 계층

- 트랜잭션 상태전이 관리
- 메시지 재전송
- 트랜잭션 타이머 관리

▶ 전송/문법 및 인코딩 계층

- 네트워크 연결 관리
- 메시지 전송
- 메시지 파싱 및 인코딩

4. 결론

본 논문에서는 SIP 프로토콜에서 정의하고 있는 계층 중 문법 및 인코딩 계층, 전송 계층, 트랜잭션 계층과 요청 메시지의 전송을 위해 반드시 필요한 RFC3263에서 기술하고 있는 바를 하나의 모듈로 묶어 구현하였다.

SIP 공통모듈을 이용하여 다양한 SIP 애플리케이션을 신속하게 구현할 수 있을 것으로 기대된다. 특히 트랜잭션 계층은 트랜잭션 사용자와 달리 Invite 트랜잭션과 Non-Invite 트랜잭션으로만 그 처리가 구분이 되고[2,3], 전송 계층은 RFC3263에서 기술하는 데로 처리되기 때문에 쏟아져 나오는 다양한 SIP 확장(Extensions) 또한 쉽게 이루어질 수 있다.

향후 본 논문에서 구현한 SIP 공통모듈을 기반으로 SIP VoIP 애플리케이션을 구현할 예정이며, 인스턴스 메시징, 컨퍼런스 등의 애플리케이션을 개발하는데 필요한 SIP 확장을 추가해 나갈 예정이다.

5. 참고문헌

[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