SIP(Session Initiation Protocol) 스택의 설계 및 구현

노강래° 이종열 김준일 신동일 신동규
세종대학교 컴퓨터공학과
{krnoh, leemaster, junil, shindk, dshin}@gce.sejong.ac.kr

Design and Implementation of a SIP Stack

Kang-rae Noh° Jong-youl Lee, Jun-il Kim, Dong-il Shin, Dong-kyoo Shin
Dept. of Computer Engineering, Sejong University

요 약

SIP는 인터넷 텔레포니 신호(Internet Telephony Call)와 같은 멀티미디어 세션을 성립, 변경, 종료시킬 수 있는 응용계층의 호 제어 프로토콜이다. 또한, SIP 프록시 서버(Proxy Server)라는 네트웍 호스트로 구축된 인프라를 통해 멀티미디어 통신을 원하는 네트웍상의 사용자를 찾아 멀티미디어 세션(Session)이 성립될 수 있도록 도와준다. SIP는 Request-Response 방식의 프로토콜이기 때문에 요청(Request)과 응답(Response)을 주고받는 일련의 트랜잭션(Transaction)으로 하나의 Task를 완성한다. 본 논문에서는 rfc2543을 기반으로 기본적인 SIP 트랜잭션을 위한 SIP 스택(Stack)의 설계 및 그 구현 방안을 제시한다.

1. 서 론

인터넷상에서 운용되는 대다수의 IP 텔레포니 관련 응용프로그램들은 인터넷 커뮤니케이션을 원하는 참여집단 사이의 데이터 교환을 성립시키는 세션의 생성과 관리를 필요로 하고 있다.[1] 이러한 세션의 관리는 참여자들의 행동양식으로 인해 그 구현이 매우 복잡하다. 가령, 사용자가 말단(End-point)사이를 이동하거나, 인터넷상에서의 자신을 식별할 수 있는 ID를 여러 개 가지고 있을 수 있으며, 상이한 미디어 포맷을 이용하여 참여자간의 멀티미디어 커뮤니케이션을 시도 할 수 도 있다. 이러한 구현상의 어려움을 SIP[2]는 "User Agent" 라는 인터넷 말단을 이용하여 해결한다. User Agent 는 프록시 서버라는 네트웍 호스트들로 이루어진 인프라를 통해 참여자의 위치 추적 및 기타 서비스를 위한 요청을 보낼 수 있으며, 참여자간의 복잡한 세션의 관리를 용이하게 해준다.[3] SIP 메시지는 SIP 스택을 통해서 처리되거나, 새롭게 구성되어 전송된다. 본 논문에서는 서론에 이어 2장의 관련연구를 통해 SIP의 특성을 설명한다. 3장에서는 본 논문이 제시한 SIP 스택의 구조적 특징을 설명하고, 4장에서는 SIP 스택의 설계 및 각 모듈의 기능을 설명한다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대하여 설명한다.

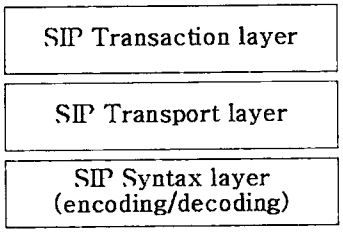
2. 관련연구

SIP는 웹 기반의 프로토콜로 HTTP(Hyper Text Transfer Protocol)및 SMTP(Simple Mail Transport Protocol)와 유사한 프로토콜이며, 다음과 같은 특성을 가지고 있다. 첫째, SIP는 수직적으로 통합된 커뮤니케이션 시스템이 아니라, 거의 Component에 가까운 구조를 가지고 있다. 즉, 보다 완벽한 멀티미디어 구조를 이루기 위해 RTP(Realtime Transfer Protocol), RTSP(Realtime Transfer Streaming Protocol), MGCP(Media Gateway Control Protocol) 와 같은 IETF(Internet Engineering Task Force)[4]프로토콜들을 사용 할 수 있으며, 상기 프로토콜에 독립적으로 작용한다. 둘째, SIP는 서비스자체를 제공하지 않으며, 단지 서비스를 위한 근본적인 것을 제공한다. 예로, SIP는 SIP 호스트로 구성된 SIP 네트웍 상의 사용자의 위치를 찾을 수 있으며, 해당 사용자에게 어떤 최소한의 정보를 전달 할 수 있다. 만일 그 최소한의 정보가 세션에 대한 정보(가령, SDP[5]: Session Description Protocol)가 들어가면, 말단사이의 세션 파라메터를 일치시킬 수 있으며, 동시에 멀티미디어 통신이 가능해진다. 셋째, SIP에 의해 제공되는 서비스의 본질상 보안은 상당히 중요한 사항이 된다. 따라서 SIP는 Dos(Denial of service) 공격으로부터 의 보호, SIP 서비스에 대한 인증, 메시지의 암호화, 개인 정보를 위한 보안 정책 등을 포함한 보안 서비스를 제공한다. 마지막으로, SIP는 IPv4, IPv6 모두에서 사용 가능하다.

3. SIP 스택의 구성

본 장에서는 SIP 스택의 각 계층별 특징을 설명한다.

본 논문이 제시한 스택은 [그림1]과 같은 계층화된 구조를 가지고 있다. 각 계층 사이의 결합력(coupling)을 낮게 설계했기 때문에 계층별로 독립적인 처리가 가능하다.



[그림1] SIP 스택의 구조

3.1 SIP Syntax layer (encoding/decoding)

SIP 구문을 구성(encoding)하고, 처리(decoding)하는 계층이다. SIP 구문 계층(Syntax layer)은 SIP 메시지를 구성하는 헤더 정보를 의미한다. 이 계층은 SIP 트랜잭션 계층(Transaction layer)에 의해 동적으로 사용된다.

3.2 SIP Transport layer

전송계층(Transport layer)은 클라이언트에게 요청 및 응답메시지를 어떻게 보내고 받는지가 정의 되어있다. SIP 의 모든 요소들은 이러한 전송계층을 가지고 있다.

3.3 SIP Transaction layer

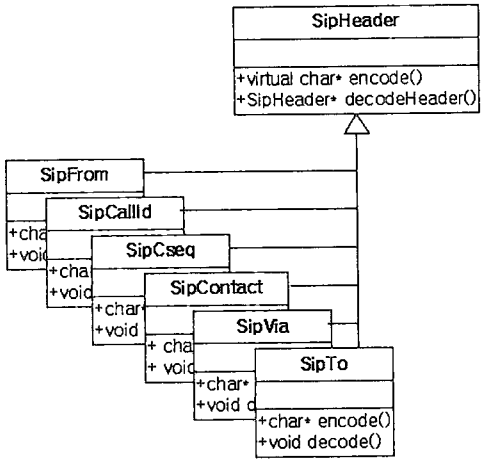
트랜잭션 계층은 SIP 스택의 근간이 되는 계층이다. SIP 트랜잭션은 클라이언트와 서버사이에서 보내진 요청과 보낸 요청에 대한 응답 모두를 의미한다. 트랜잭션 계층은 메시지의 재전송, 요청 메시지의 구성 및 요청에 따른 응답 메시지의 구성 등을 담당한다.

4. SIP 스택의 설계 및 구현

본 논문에서 제시한 SIP 스택은 C++로 구현하였다. SIP 스택의 테스트는 VOVIDA[6]에서 오픈 소스로 제공하는 User Agent에 본 논문이 제시한 SIP 스택을 붙여 컴파일 한 후, UA(User Agent)to-UA 트랜잭션에 제한을 두어 테스트를 하였다. 본 장에서는 SIP 스택을 구성하고 있는 각 계층별 클래스 구조 및 각 모듈의 기능을 설명한다.

4.1 SIP Syntax 처리 모듈

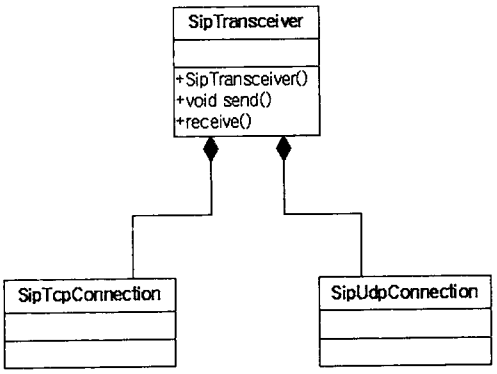
[그림2]에서 각 헤더 클래스들은 SIP 헤더 필드를 구성하고, 처리하는 멤버 함수가 정의 되어있다. 이들 클래스는 SipHeader 클래스를 상속받음으로써 SipHeader 객체를 통해 각 헤더의 객체들로 접근 할 수 있도록 설계하였다. 헤더 클래스들은 encode, decode 함수와 더불어 헤더 파라메터를 멤버 변수로 가지고 있다. decode 함수는 헤더 필드를 구성하고 있는 각 요소를 분리하여, 각 클래스의 멤버 변수에 그 값을 저장한다. encode 함수는 각 헤더 클래스들이 가지고 있는 멤버 변수를 이용하여 하나의 헤더 라인을 구성한다.



[그림2] SIP 구문 처리 모듈의 클래스 계층도

4.2 SIP Transport 처리 모듈

트랜잭션에 의해 만들어진 SIP 메시지를 목적지로 보내거나 목적지에서 보낸 SIP 메시지를 받는 모듈이다.



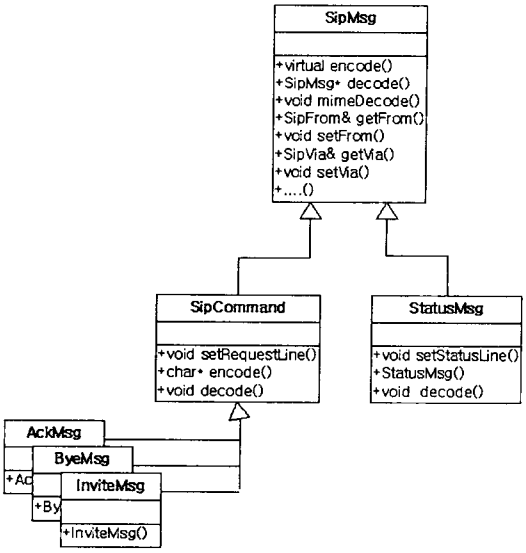
[그림3] SIP 전송 처리 모듈의 클래스 계층도

[그림3]의 SipTranscevier 클래스는 SipTcpConnection 클래스 또는 SipUdpConnection 클래스의 생성자를 통해 TCP 또는 UDP 커넥션을 만든다. 이후, 메시지의 송신은 send 함수를 이용하고, 수신은 receive 함수를 이용한다. receive 함수는 제한 시간을 인자로 받아서, 제한 시간동안 SIP 메시지를 기다리며 대기 상태가 된다. SIP 전송 모듈은 SIP 프록시 서버와 User Agent에서 사용된다.

4.3 SIP Transaction 처리 모듈

SipMsg 클래스는 트랜잭션에 따른 SIP 메시지를 구성하거나 전송계층을 통해 들어온 SIP 메시지를 처리하여, 각 헤더 필드 타입의 헤더 정보를 가져올 수 있는 멤버 함수들로 구성되어있다. SipCommand 클래스의 하위 클

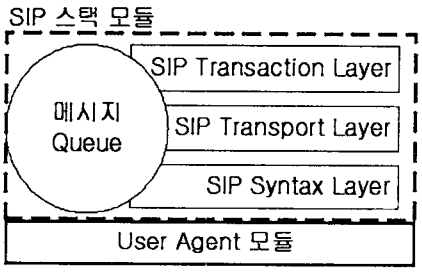
래스인 각 SIP 메소드(Method) 클래스는 SIP 구문 계층을 구성하고 있는 헤더 클래스들을 이용하여 전체 SIP 메시지 중 헤더 부분을 구성한다.



[그림4] SIP 트랜잭션 처리 모듈의 클래스 계층도

[그림4]에서 SipCommand 클래스는 하위 클래스로부터 구성된 헤더 부분에 Request-Line을 덧붙여, 실제적인 요청 메시지를 구성한다. 트랜잭션 처리 모듈은 User Agent에서 사용될 수 있다. User Agent는 일련의 연속적인 트랜잭션을 수행함으로써 하나의 Task를 완성한다.

4.4 SIP Stack 테스트용 UserAgent 모듈



[그림5] 테스트용 User Agent 모듈의 구조도

테스트용 User Agent는 본 논문이 제시한 SIP 스택을 테스트하기 위한 모듈이다. SIP 각 계층은 SIP 메시지 큐(Queue)를 서로 공유한다. 전송 계층을 통해 들어온 SIP 메시지는 메시지 큐에 저장되고, 트랜잭션 계층을 통해 큐에 저장된 SIP 메시지가 처리된다. 하위의 User Agent 모듈은 SIP 시그널로 컨트롤 할 수 있는 미디어 프로토콜과의 중간 인터페이스 역할을 담당한다. 그림에는 없지만, RTP나 RTSP와 같은 미디어 프로토콜 처리 모듈이 올 수 있다.

5. 결론 및 향후 연구 과제

SIP는 그 구현 방향이 rfc2543 문서에 명세 되어 있다. IETF 는 2002년 2월에 rfc2543 bis 버전 7을 발표했으며, 현재 계속 발전 진행중이다. 결국, SIP 스택의 설계는 구조상 확장성을 가져야 하며, 수정 및 보완이 용이한 구조를 가지고 있어야 한다. 본 논문이 제시한 SIP 스택은 스택의 구조를 3개의 계층(SIP Syntax layer, SIP Transport layer, SIP Transaction layer)으로 나누어 각 계층별 고유한 기능을 독립적으로 수행한다. 따라서 구현된 스택의 기능상의 변경 및 그 확장이 용이한 장점을 가지고 있다. 향후 연구 과제로는 현재 구현된 SIP 스택의 기능의 확장과 이를 이용하여, User Agent 와 프록시 서버의 설계 및 구현이다.

참고문헌

[1]Alan B.Johnston "SIP: Understanding the Session Initiation Protocol"
[2]SIP(Session Initiation Protocol) http://www.cs.columbia.edu/sip/
[3]Rosenberg,Schulzrinne,Camarillo,Johnston,Peterson,Sparks,Handly,"SIP: Session Initiation Protocol", Request for Comments 2543, Internet Engineering Task Force, February 4, 2002
[4]IETF(Internet Engineering Task Force) http://www.ietf.org/
[5]M.Handley,V.Jacobson "SDP: Session Description Protocol" April,1998
[6]Vovida: http://www.vovida.org