

Hybrid형 SIP 서버를 통한 인터넷폰 설계

김진수*, 양해권**

*군산대학교 대학원 정보통신공학과, **군산대학교 정보통신공학과 교수

Design of Internet Telephone by SIP Server of Hybrid Type

Jin-Soo Kim*, Hae-kwon Yang**

Dept. of Telecommunications, Kunsan National University

요 약

인터넷을 이용한 여러 응용 서비스들 중에서 저렴한 비용으로 음성을 전송할 수 있는 VoIP 서비스의 발전으로 사용자의 급격한 증가가 예상된다. VoIP에 mobility, universal number, multiparty conference, voice mail, automatic call distribution과 같은 고품질의 서비스를 제공하기 위해서는 시그널링이 가능한 표준화된 프로토콜이 필요하다. 현재 IETF의 SIP(Session Initiation Protocol)가 빠른 호 설정과 parsing 및 compile이 쉬운 장점으로 인해 SIP를 기반으로 한 VoIP 서비스를 제공하기 위해 국내외적으로 SIP 기반 구성요소에 대한 개발이 진행 중이다. 본 논문에서는 사용자가 보내는 request(INVITE) 메소드(method)를 처리해주는 SIP 서버의 부하 경감, 망 운용의 효율성, 많은 사용자에게 대한 서비스를 제공하기 위해 새로운 서버 유형인 Hybrid형 SIP 서버를 제안하고, 이를 이용하여 새로운 타입의 인터넷폰을 설계하였다.

1. 서 론

인터넷을 통하여 기존 전화망의 음성 서비스를 제공하는 기술이 VoIP 기술이다. VoIP는 수많은 인터넷 사용자들로부터 음성 서비스뿐만 아니라, 그 외 부가 서비스를 쉽게 제공해줄 수 있어 많은 관심을 받고 있다.

VoIP 서비스를 제공하기 위해서는 상대방(callee)을 찾아 시그널링하는 기술이 필요한데, VoIP 시그널링 기술로는 ITU-T의 H.323과 IETF의 SIP가 있다. 얼마 전까지만 해도 H.323 기반의 VoIP 서비스 개발이 많이 이루어졌으나, 최근에 대두된 SIP는 빠른 호 설정과 parsing 및 compile이 쉽고 확장성이 뛰어나며, 텍스트 기반이기 때문에 H.323에 비해 구현이 매우 쉽다는 특징을 가지고 있어 차세대 VoIP 표준 기술로 대두되고 있다.[1]

본 논문에서는 차세대 VoIP 표준기술로 자리 매김하고 있는 SIP 기술요소 중 사용자의 호 연결 요청을 받아 처리해 주는 구성요소인 Proxy/Redirect 서버에 대해 연구하여, 기존의 서로 분리된 Proxy/Redirect 서버보다 여러 면에서 우위를 가지는 Hybrid형 SIP 서버를 제안하고 이를 이용하여 인터넷 폰을 설계하고자 한다. 본 논문의 구성은 서론에 이어서 2장에서는 SIP에 대해 소개하고, 3장에서는 Hybrid형 SIP 서버

설계에 대해 기술하며, 4장에서 UA(User Agent)와 SIP 서버의 구성 및 동작을 살펴본 후 마지막 5장 결론을 맺는다.

2. SIP(Session Initiation Protocol)

2.1 SIP 개요

SIP는 사용자 단말간에 멀티미디어 세션을 생성, 수정, 해제하는 응용 계층의 호 시그널링 프로토콜이다. 멀티미디어 세션에는 컨퍼런스, 인터넷 전화, 원격 교육, 메시지 전송 등이 포함된다. SIP는 HTTP(Hypertext Transfer Protocol), e-mail, SMTP(Simple Mail Transfer Protocol), 웹에 기반을 두고 모델링되었다.

그리고 기본적으로 클라이언트와 서버간에 Request와 Response 메시지를 주고받는 클라이언트-서버 프로토콜이라고 할 수 있다. SIP는 HTTP와 유사하며, 확장성이 용이하고, 사용자의 이동성을 제공한다. UDP 및 TCP에 무관하게 동작하기 때문에 하부계층에 독립적이며, SDP(Session Description Protocol)를 세션의 정보를 협상하기 위한 메시지 바디로 사용한다.[2]

2.2 SIP 메시지

SIP는 HTTP에 기반을 두고 설계된 프로토콜이기 때문에 프로토콜의 메시지는 텍스트로 구성된다. 메시지는 헤더와 메시지 바디로 구성되는데, 헤더는 SIP 제어정보를 포함하며, 바디는 호 설정시 음성 및 비디오 코덱과 같은 양측의 능력을 협상하기 위한 정보를 포함하며 SDP로 기술된다. SIP 메시지는 클라이언트에서 서버로 보내는 Request와 서버에서 클라이언트로 보내는 Response가 있다. SIP Request는 INVITE, ACK, BYE, CANCEL, REGISTER, OPTION, INFO의 7가지 메소드로 구성된다.

INVITE는 클라이언트와 서버간에 호를 개시하는 가장 기본이 되는 메소드로 사용자와 서비스가 세션에 참가하도록 하며, 발신자와 수신자의 주소, 호의 주제, 호의 priority, 호 라우팅 Request, 바람직한 Response 특성을 포함한다. 그리고 ACK는 호 설정을 확인하고, CANCEL은 호 설정 도중에 호를 취소하며, BYE는 호 설정을 종료하고, REGISTER는 사용자의 위치를 등록하고, OPTION은 호 설정 없이 상대방의 기능을 파악한다.[2] 그림 2.1에서 간단한 SIP 메시지 구조를 보이고 있다.

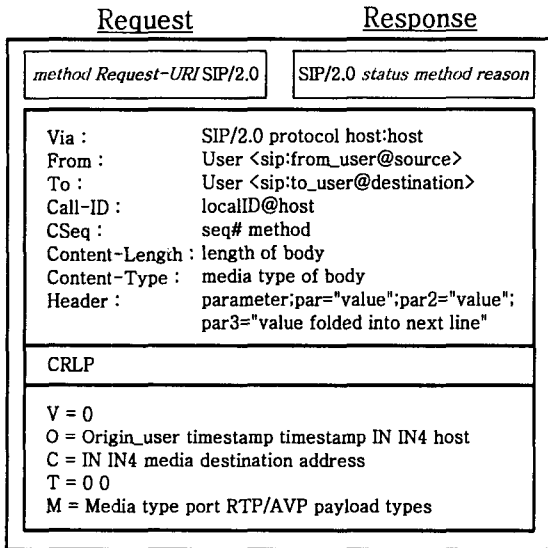


그림 2.1 SIP 메시지 구조

수신자는 Request 메시지를 받고 나면 SIP Response 메시지로 응답을 해야한다. Response의 상태코드는 HTTP/1.1 Response code와 유사하나 확장된 형태이며, 모든 HTTP/1.1 Response code가 그대로 적용되는 것은 아니다. 표 1에서 SIP Response의 상태코드를 보여주고 있다.

표 1. SIP Response Code

메시지	의미	상태
1xx	Informational Messages	Provisional
2xx	Successful Responses	Final
3xx	Redirection Responses	Final
4xx	Request Failure Responses	Final
5xx	Server Failure Responses	Final
6xx	Global Failures Responses	Final

2.3 SIP 구성 컴포넌트

SIP는 크게 UA(User Agent)와 SIP 서버로 구성된다. UA는 사용자가 호를 연결할 수 있도록 지원되는 단말 시스템으로, caller의 기능을 수행하는 UAC(User Agent Client)와 callee의 기능을 수행하는 UAS(User Agent Server)로 분류할 수 있다. 호 연결에 있어 UAC는 사용자의 호 연결 요청을 개시하기 위해 사용되고, UAS는 요청을 수신하고 사용자의 입장에서 응답을 보내는 기능을 수행하는 단말이다.

SIP 서버는 Proxy, Redirect, Registrar 서버로 구성된다. Proxy 서버는 UAC로부터 호 연결 요청을 받게 되면, callee의 위치 정보를 Location Services를 해주는 Registrar 서버에게 contact하여 callee의 정보를 얻게된다. 그런 다음 callee의 위치 정보를 가지고 UAC가 보낸 INVITE 메소드를 재조립한 다음 파악된 위치 정보 상의 서버에게 전달함으로써, UAC와 UAS 기능을 수행한다. 즉 UAC의 요청에 대한 처리를 스스로 수행하며 서버 자신이 요청 메시지를 생성하여 목적지 UAS에 전송할 수 있는 기능을 갖고 있다. Proxy 서버는 동작 방식에 따라 stateful과 stateless로 나뉜다. stateful은 각각 유입된 호에 대한 정보를 기억하며 이것으로 호를 생성하고 처리하는 반면, stateless은 각 호를 처리하고 그 호에 대한 상태 정보를 기억하지 않는다.

Redirect 서버는 UAC의 SIP 요청메시지를 수신하면 상대방 UAS와 연결하기 위한 서버의 주소를 UAC에게 알려줌으로써 UAC가 직접 다음 서버에 요청메시지를 보낼 수 있도록 한다. 즉 Redirect 서버는 사용자의 요청에 대한 응답 기능만을 수행할 뿐 스스로 요청 메시지를 생성하지 않는다. 그러므로 Proxy 서버에 비해 수행해야하는 작업이 적음으로 서버의 부담을 최소화할 수 있다는 장점을 가지고 있다.

마지막으로 Registrar 서버는 각 SIP 서버의 위치를 알려주는 기능과 사용자의 위치 정보를 등록하고 수정 및 삭제 기능을 수행한다. 이러한 등록 서버의 이용으로 각 서버는 UAC로부터의 요청에 대한 실제 목적지 서버의 위치 및 사용자의 위치 정보를 파악할 수 있게 된다. SIP Registrar 서버는 Proxy, Redirect 서버와

상호 연동하며 Location Services를 제공한다.[2] [3]

2.4 SIP 동작

SIP 동작은 입력 Request를 처리하는 방법에 따라 Proxy 모드와 Redirect 모드로 나눌 수 있다.

그림 2.2에서 Proxy 서버는 INVITE Request를 받아 그 Request의 어드레스를 보고 Registrar 서버에게 contact하여 callee의 정확한 위치 정보를 얻는다. 그런 다음 Proxy 서버는 Registrar 서버에게 받은 어드레스로 SIP Request를 보내고, Request를 받은 UAS는 callee에게 INVITE 메시지가 왔음을 알리고 Proxy 서버에게 메시지를 잘 받았다는 응답을 보낸다. 그러면 Proxy 서버는 UAC에게 OK 응답을 보내고, UAC는 Proxy 서버에게 ACK 메시지를 보내며 Proxy 서버는 UAS에게 ACK 메시지를 포워딩함으로써 메시지 전송이 성공적으로 이루어짐을 알리게 되고 호가 설정된다. 여기서 ACK는 Proxy 서버를 거치지 않고 수신자에게 직접 보내질 수도 있다.

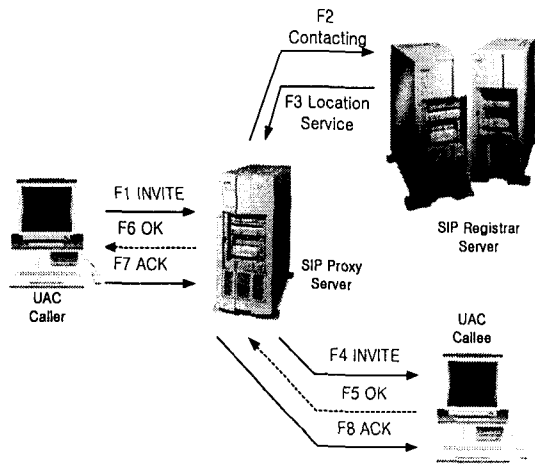


그림 2.2 Proxy 모드

그림 2.3에서 Redirect 서버는 INVITE Request를 받아 Proxy 서버처럼 Request의 어드레스를 보고 Registrar 서버를 contact하여 callee의 정확한 위치 정보를 얻는다. 하지만 Redirect 서버는 새로 얻어진 어드레스로 callee와 직접 연결을 시도하지 않고 UAC에게 그 어드레스를 되돌려 주게되고, UAC가 서버에게 되돌려 받은 어드레스로 새로운 Request를 보내게 된다. 그리고 호 요청이 성공적으로 이루어지면 UAC와 UAS는 ACK를 주고받음으로써 호가 설정된다.[2] [4]

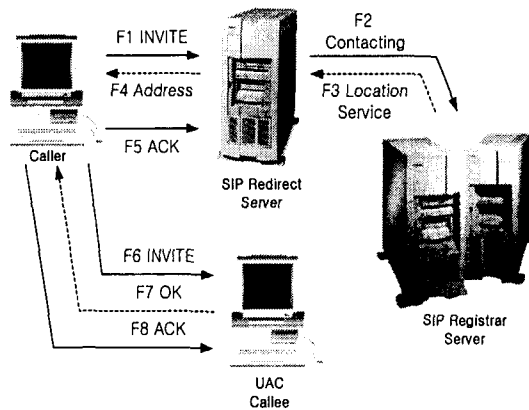


그림 2.3 Redirect 모드

3. SIP Proxy/Redirect 서버 설계

3.1 Hybrid형 SIP 서버 제안

2장에서 기술한바와 같이 호 설정 요구의 처리방식에 따라 Proxy 모드와 Redirect 모드로 나뉘어진다. 즉 Proxy 모드로 동작하면 Redirect 모드에 비교하여 직접 callee에게 INVITE Request를 보냄으로써 호 설정이 더 빨리 이루어질 수 있다. 하지만 Registrar 서버로부터 받은 callee의 위치 정보를 가지고 INVITE Request를 다시 형성하여 보내기 때문에 많은 사용자들이 호 연결 요청을 시도하면 Proxy 서버의 부하와 처리 지연시간이 증가하는 단점이 발생된다. 그러므로 네트워크 구조가 복잡하거나 방대한 경우 Proxy 서버만으로는 UAC들의 요청을 수용할 수 없을뿐더러 사용자에게 대한 서비스율도 떨어지게 된다. 이에반해 Redirect 모드는 Proxy 모드에 비교하여 Registrar 서버로부터 받은 callee의 위치 정보를 caller에게 보내주게 되고 caller가 callee에게 직접 INVITE Request 메시지를 보냄으로써 호가 성립되는 방식으로 caller와 callee간에 호 설정에 지연이 발생한다는 단점이 있지만, Proxy 서버에 비하여 상당히 적은 부하를 갖게 되어 적은 처리 지연시간으로 많은 사용자들에게 서비스할 수 있다. 즉 Registrar 서버는 Proxy 서버보다 호 연결에 있어 적은 처리를 하는 동작 방식으로 보다 많은 UAC의 호 요청을 수용할 수 있다.

그렇다면 망 운영자는 어떠한 동작 모드를 선택할 것인지를 신중히 고려해야 한다. 즉 서버의 부하를 줄이고, 많은 사용자들의 요구를 처리하기 위해 Redirect 모드 방식을 선택할 것인지, 아니면 서버의 부하는 가중되더라도 빠른 호 설정과 부가 서비스를 사용자들에게 제공해주기 위해서 Proxy 모드를 선택할 것인지를 고려해야 한다. 물론 Proxy 모드 방식이 여러 가지 즉

면에서 Redirect 모드에 비해 상당히 효율적임은 명확하다. 하지만 그에 따르는 단점을 고려하지 않을 수는 없다. 그래서 본고에서는 망 운영의 효율성과 서버의 부하 경감, 더 많은 사용자들에 대한 서비스, 가능한 사용자들에게 최선의 서비스를 제공해줄 수 있는 새로운 SIP 서버를 제안하게 되었다.

3.2 Block 구성도 및 Operation

그림 3.1에서 제안한 서버의 구성도를 볼 수 있다. Proxy 모드로써 동작할 수 있는 처리 모듈과 Redirect 모드로써 동작할 수 있는 처리 모듈을 함께 탑재함으로써 두 모드를 동시에 처리할 수 있는 방식이다.

우선 *preProcessor*는 사용자들의 INVITE Request를 수신하게 되면 Proxy/Redirect 모드 방식 중 어느 모드로 동작할 것인지를 결정한다. 결정 기준은 Proxy 서버 모듈이 가지고 있는 큐의 상태를 보고 결정을 하게 된다. Proxy 서버 모듈의 큐는 두 개의 임계치를 가지고 있는데 Proxy 모드로 동작할 수 있는 한계를 나타내는 *th2*와 Redirect 모드에서 Proxy 모드로 전환할 수 있는 결정 기준이 되는 *th1*이다.

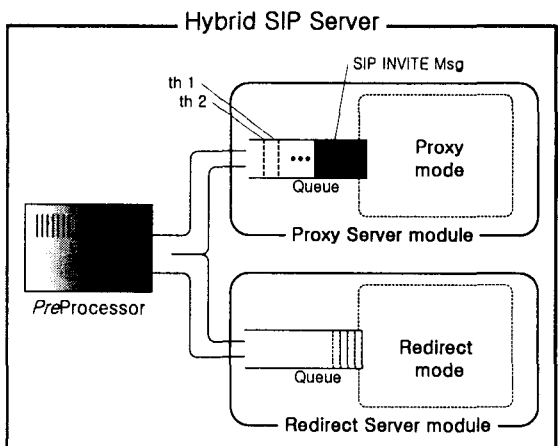


그림 3.1 Hybrid형 SIP 서버의 구성도

사용자들이 보낸 호 설정 요구 메시지를 받게되면 두 번째 임계치인 *th2*를 넘지 않는 한 Proxy 서버 모듈의 큐로 메시지를 보냄으로써 Proxy 모드로 동작하게 한다.

만약 사용자로부터 호 요청을 받았을 때 Proxy 서버 모듈의 큐가 두 번째 임계치인 *th2*에 도달하게 된다는 것을 감지하게 되면 *preProcessor*는 그 시점부터는 Redirect 서버 모듈의 큐로 호 설정 요구 메시지를 보내 주게 되어 Redirect 모드로 동작함으로써 그동안 Proxy 서버의 부하를 줄일 수 있게 된다. 호 연결 속도 면에서는 Proxy 모드 방식보다는 약간 늦어지지만

Proxy 모드 방식으로만 동작하는 기존 Proxy 서버의 부하를 경감하는 결과를 가져올 수 있다.

사용자들로부터 INVITE Request 메시지를 받게 되면 계속해서 *preProcessor*는 Proxy 서버 모듈 큐의 임계치를 조사하여 *th2*에서 *th1*으로 메시지가 줄어들게 되면 그 이후에 도착한 메시지들에 대해서는 Proxy 서버 모듈 큐로 메시지를 보내주게 됨으로써 Proxy 모드의 효율성을 최대한 활용하는 측면과 SIP 서버의 부하 경감 측면을 동시에 고려한 서버를 이용하여 인터넷 폰을 설계하고자 한다.

3.3 preProcessor의 알고리즘

서버의 동작 모드를 결정하는 *preProcessor*의 알고리즘이 그림 3.2에 나타나 있다.

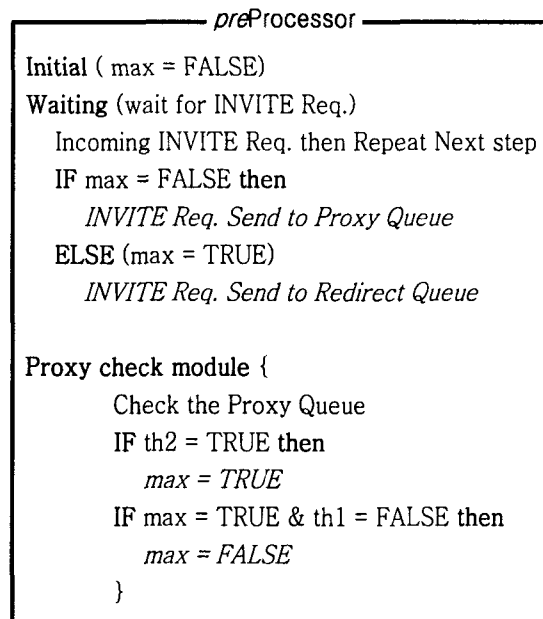


그림 3.2 preProcessor의 알고리즘

사용자들의 INVITE Request를 기다리고 있는 Waiting 상태에서 메시지가 도착하면 *max* 값에 따라 Proxy 모드로 동작할 것인지 아니면 Redirect 모드로 동작할 것인지를 결정한다. 즉 Proxy 서버 큐의 한계치를 나타내는 *max*값이 FALSE 동안에 Proxy 큐로 INVITE Request를 보냄으로써 Proxy 모드로 동작할 수 있게 된다. Proxy 체크 모듈에서는 *th2*값이 TRUE가 되면 *max*값을 TRUE로 설정하게 되고, *preProcessor*는 *max*값이 TRUE인 동안에는 INVITE Request는 Redirect 큐로 보내주게 되어 Redirect 모드로 동작하게 한다.

Proxy 체크 모듈은 계속해서 Proxy 큐를 검사하여

큐에 저장된 메시지 양이 th1까지 감소하게 되면 max 값을 FALSE값으로 설정하여 그 이후에 도착한 메시지에 대해서 preProcessor는 Proxy 큐로 INVITE Request를 보내어 Proxy 모드로 동작할 수 있게 된다. 알고리즘에서 보는 바와 같이 preProcessor의 동작은 상당히 간단하다. 즉 Proxy 체크 모듈이 처리하는 max 값에 따라 INVITE Request를 해당 큐로 보냄으로써 그 동작이 수행된다. 일반적인 서버유형에 preProcessor가 추가된 Hybrid SIP 서버는 preProcessor가 간단한 처리동작을 수행하기 때문에 부하를 더 과중시키는 일은 발생하지 않는다.

Proxy/Redirect 서버 모듈은 사용자의 INVITE Request를 처리하기 위해서는 Registrar 서버에게 callee의 위치 정보를 파악하기 위하여 Registrar 서버에 contact한 후 각각의 모드에 따라 동작하게 된다.

4. UA의 구성 및 동작

4.1 UAC(User Agent Client)

실시간 응용 프로그램에 있어 지연은 극복해야 할 가장 커다란 문제라고 할 수 있다. 지연의 문제를 해결하기 위해 현재 RTP/RTCP가 인터넷을 통한 실시간 데이터 전송 서비스에서 거의 표준처럼 사용되고 있는 실정이다.

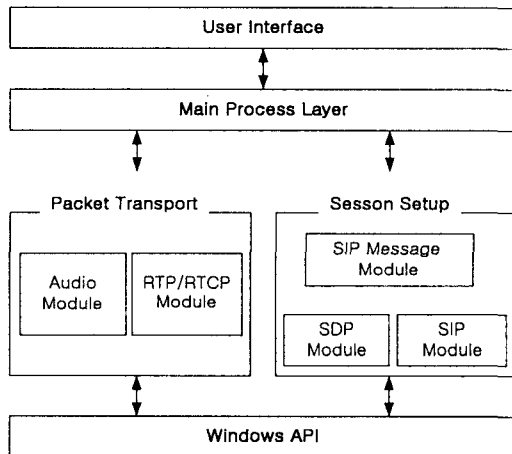


그림 4.1 UAC의 구성

기본적으로 UAC는 Audio 모듈, RTP/RTCP 모듈, SIP 메시지 모듈이 존재한다. SIP 모듈은 사용자의 요청에 따라 SIP 메시지를 생성하고 처리하여 호 연결이 가능하도록 한다. SIP 모듈에 의해 SIP 호 연결이 이루어지면 음성 패킷이 교환될 수 있는 세션이 개설되고 사용자의 음성은 호 연결시에 협상된 코덱

으로 변환되고, RTP 모듈을 통해서 상대방으로 전달하게 된다.

4.2 SIP 서버

기본적인 SIP 서버는 UAC들간의 시그널링만을 담당하므로 서버의 성능과 안정성이 요구된다. 그림 4.2에서 제안한 SIP 서버의 전체적인 구성을 보여주고 있다. 우선 Socket API로부터 받은 SIP 메시지는 preProcessor를 통해 현재 서버의 상태에 알맞게 Proxy 혹은 Redirect 모듈로 분배되게 된다. 각 모듈에서는 각각의 처리 방식에 따라 메시지가 분석 및 처리되며 마지막으로 Registra 서버를 통해 Location Services를 받게 된다.

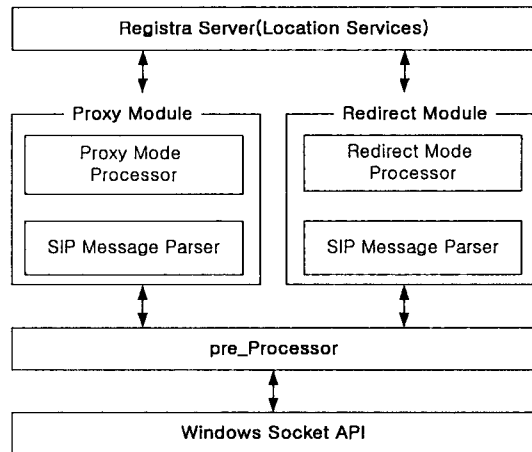


그림 4.2 SIP 서버의 구성

UA와 SIP 서버의 동작측면을 살펴보면 SIP 서버가 새롭게 제안되었다 해서 UAC와 SIP 서버와의 동작에는 변화될게 없으므로 앞에서 설명했던 그림 2.2의 Proxy 모드 방식이나 2.3의 Redirect 모드 방식 중 preProcessor의 결정에 따라 동작하게 된다.

5. 결 론

현재 국내·외적으로 각광을 받고 있는 인터넷 멀티미디어 통신시대를 지향하는 기술중의 하나인 VoIP 기술은 앞으로도 계속해서 많은 발전이 있을 것이다. VoIP 기술을 지원하기 위해 기본적으로 필요한 호 처리를 위한 프로토콜로 현재 SIP가 표준으로 자리 매김하고 있으며, SIP 프로토콜을 지원할 수 있는 제품 개발에 여러 회사에서 개발 중이다.

본 논문에서는 사용자의 요청에 보다 많은 부가 서비스의 지원과 호 설정이 빠르게 이루어지는 Proxy

모드의 장점을 최대한 살리고, 많은 사용자들의 호 처리 요청시 서버의 부하를 경감하기 위해 Redirect 모드로의 전환을 피할 수 있는 서버를 제안하여 인터넷 폰을 설계하였다. Hybrid형 SIP 서버는 다양한 네트워크에 쉽게 적용될 수 있는 특징을 가지고 있다. 결과적으로 효과적인 사용자 서비스 지원과 능동적으로 망과 결합할 수 있음으로써 커다란 기대효과를 가져올 수 있을 것이다.

참고문헌

- [1] [http://www.cs.columbia.edu/~hgs/sip/drafts/sip_h323v4 .doc](http://www.cs.columbia.edu/~hgs/sip/drafts/sip_h323v4.doc)
- [2] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: session initiation protocol", draft-ietf-sip-rfc2543bis-05.txt, Oct. 2001.
- [3] H. Schulzrinne, J. Rosenberg, "SIP - Location SIP Servers", draft-ietf-sip-srv-02.txt, Mar. 2001.
- [4] J. Rosenberg, K. Summers, H. Schulzrinne, "SIP Call Flow Examples", draft-ietf-sip-call-flows-05.txt, Jun. 2001.