

CPL 처리를 위한 SIP 프락시 서버와 CPL 서버간 동작 메카니즘

*민경주, 이종화, 강신각, 박기식

{meanrace, jhyiee, sgkang, kipark}@etri.re.kr

* 한국전자통신연구원

Behavior Mechanism Between SIP Proxy Server And CPL Server for CPL Processing

*Kyoung Ju Min, Jong Hwa Yi, *Shin Gak Kang, *Ki Shik Park

요 약

호처리 언어(CPL : Call Processing Language)는 공중 전화망에서 제공하는 다양한 부가 서비스를 인터넷상에서 개발할 수 있는 기술로서, IETF IPTTEL 워킹그룹에서 개발한 표준이다. 사용자들은 CPL을 이용하여 호 전환과 같은 다양한 부가서비스를 요청할 수 있으며 이들은 SIP Registrar에 등록되어 실질적인 서비스를 제공받게 된다. 본 논문에서는 인터넷 텔레포니 서비스를 이용하면서 다양한 부가서비스 지원을 위한 프락시 서버와 CPL 서버 사이의 동작에 대해 프락시 서버 측면에서 기술하고자 한다. CPL을 지원하는 SIP 프락시 서버는 리눅스 커널 버전 2.4.x 상에서 C언어를 이용하여 설계 및 구현하였고, CPL 서버는 라이브러리로, 프락시 서버와 연동하여 동작한다.

1. 서론

인터넷의 급속한 확산으로 인해 IP 망의 이용이 점차로 증가되어 가고 있으며, 이러한 IP 망을 이용하여, 무료 또는 저비용의 인터넷 텔레포니 서비스를 이용하기 위한 방법으로 H.323이 주로 이용되었다. 하지만, 이러한 H.323이 주도하고 있던 인터넷 전화 시장에서 SIP가 많은 부분을 차지한 상태이다. SIP는 사용자가 현재의 위치 정보를 레지스트라에게 등록할 수 있으며, 특정 컨퍼런스 제어 프로토콜에 얽매이지 않고, 하위 계층의 프로토콜에 독립적으로 동작하여 추가 적인 기능 확장에 용이한 특징을 갖는다. 이러한 이유로 개발된 SIP에 공중전화망에서와 같은 다양한 부가 서비스를 인터넷 상에서 지원할 수 있도록 IETF IPTTEL 워킹그룹에서 개발한 표준인 CPL의 제공이 몇몇 업체에서 이루어지고 있으나, 사용자의 요구를 만족시키지 못하는 실정이다[1]. 이러한 이유로 본 논문에서는 기존 SIP와 완전한 호환을 이루고, 서비스의 극대화 방안으로 CPL을 처리할 수 있는 프락시 서버를 설계하고, 구현하였다. 본 논문에서는 CPL을 지원하는 프락시 서버와 CPL

서버사이의 연동 메커니즘에 대해 다루고자 한다. 본 논문에서 다루는 구현은 신속한 개발 환경과, 저비용의 장점을 가진 리눅스 플랫폼에서 이루어졌으며, 본 논문의 구성은 다음과 같다.

2장에서는 SIP 프락시 서버 구조, 동작 및 CPL 수용 구조를 소개한다 3장에서는 CPL을 지원하기 위한 프락시 서버와 CPL 서버의 연동에 관한 상세한 사항을 소개한다. 결론 및 향후 연구 과제는 4장에서 기술한다.

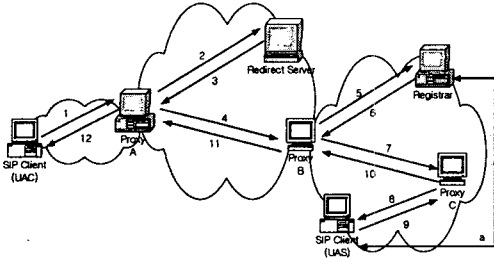
2. SIP 프락시의 구조 및 동작, CPL 수용구조

2.1 SIP 프락시의 구조

SIP의 구조는 [그림 1]의 순서에 의해 콜을 설정하게 된다. 그림에서 보는 바와 같이 프락시 서버는 아웃바운드 프락시로부터 받은 메시지를 자신이 관리하는 도메인 내의 UA인지 검사한다.

프락시는 사용자가 자신의 위치 정보를 등록한 레지스트라에게 사용자의 위치를 찾고, 받은 메시지를 Callee에게 전달하는 기능을 한다. 따라서 프락시 서버의 대표적인 기능은 UA간의 호 연결을 위해 메시지를 전달하는 것이다. 이

러한 기능을 수행하기 위해, 사용자가 등록한 위치 정보를 이용하게 된다[2, 3].

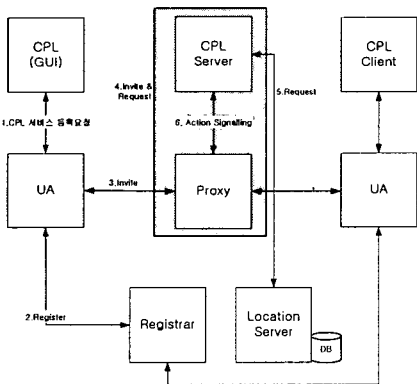


[그림 1] SIP 구조

2.2 CPL 수용 구조

프락시 서버는 가상의 UAS/UAC인 것처럼 동작한다. 루프를 방지하기 위해 프락시 서버는 수신되는 request들의 Via 헤더 필드내에 자신의 주소가 이미 들어있는지 점검해야한다. To, From, Call-ID, Contact 태그들은 원 Request로부터 정확히 복사가 된다. incoming request에 대해 루프 방지를 위한 Via 필드를 복사하고, Request URI를 보내고자 하는 곳으로 변경하고, 자신의 주소를 Via 필드에 추가하고, branch 부분을 추가한 후 보내게 된다[4,5]. 그런데 CPL을 처리하기 위해서 프락시 서버는 Request/Response에 대해 CPL 서버에게 어떠한 행동을 할 것인지에 대해 요청하게 된다. 이때 CPL 서버는 프락시의 동작을 결정한다. 이러한 CPL 서버의 지시에 따라 프락시는 리다이렉트 서버로 동작할 수도 있고, 프락시 서버로 동작할 수 있다[4,5].

3. CPL 프락시-CPL 서버의 동작



[그림 2] CPL 지원을 위한 전체 흐름도

SIP에서 CPL을 지원하기 위해서 프락시 서버와 CPL 서버 사이에 두가지 구조를 가질 수 있다. 첫째, CPL 서버가 라이브러리 형태로 프락시 서버와 동일한 머신에서 동작하는 구조이고, 둘째는 각기 별개의 서버로 동작을 하는 경우이다. 본 논문에서 구현한 CPL구조는 다음의 그림 2와 같이, 같은 머신에 CPL 서버가 모듈로 올라가 있는 형태를 갖는다.

3.1 RequestAction()

프락시는 수신한 호 설정 메시지에 대해서 CPL 서버에게 자신의 행동을 요청하게 된다. 이러한 요청을 수행하는 함수가 RequestAction()이데, 이 함수를 호출하고, 이 함수의 결과에 따른 동작 알고리즘이 그림 3에 나타나 있다.

```

if ( (ReqAction = RequestAction(pSIPMsg, &pIR))>0)
{
    switch (pIR->CPLActionType)
    {
        case NO_SCRIPT:
        case NOTHING_SPECIAL:
            break; // 기존 프락시와 동일한 동작
        case PROXY_THE_CALL:
            DoProxyTheCall(); break;
        case REDIRECT_THE_CALL:
            DoRedirectTheCall(); break;
        case REJECT_THE_CALL:
            DoRejectTheCall(); break;
        case LOOP_DETECTED:
            Send482Response(); break;
    }
} else {
    // pIR->ErrorCode에 따른 동작
}
    
```

[그림 3] CPL 서버 - 프락시 서버 간 알고리즘

이에 대해 각각을 자세히 살펴보면 다음과 같다.

NO_SCRIPTS & NOTHING_SPECIAL

NO_SCRIPTS와 NOTHING_SPECIAL은 콜리(Callee)가 등록된 스크립트가 없거나, 스크립트는 존재하지만, 현재의 INVITE에 대해서는 특별한 규약이 없는 상태를 의미한다. 이러한 경우, 프락시 서버는 기존 프락시 서버와 동일한 과정을 수행하게 된다. 즉, 콜리에 해당하는 위치정보를 로케이션 서버에서 얻어와 각각의 브랜치에 대해 포킹을 수행한다[4].

PROXY_THE_CALL

PROXY_THE_CALL은 CPL 서버에게 요청할 때 두번째 인자로 사용한 InterpreterResult_t 구조체의 위치정보를 포함

하는 LocationSet_1으로 프락시를 하게 된다. 이때 Caller는 Callee1에게 전화를 하려고 하지만, Callee1은 자신에게 오는 콜에 대해, Callee2에게 호 전환하도록 로케이션 서버에 등록한 경우이다. 이 경우 프락시 서버는 SIP 메시지의 StartLine에 있는 RequestURI를 Callee2로 바꾸고 Callee2에게 INVITE를 전달하게 된다[4].

REDIRECT_THE_CALL

REDIRECT_THE_CALL은 CPL 서버가 일시적으로 리다이렉트 서버로 동작하도록, 프락시 서버에게 명령하는 것이다. 이 경우 프락시 서버는 302(Moved Temporarily) 메시지를 Caller에게 전달하고 트랜잭션을 종료하게 된다. 이때 Caller에게 전달되는 메시지에는 pIR->pLocation이 contact 헤더에 첨가된다. 이 메시지를 받은 Caller는 contact 헤더의 주소를 RequestURI에 넣어 다시 콜 연결을 시도한다[4].

REJECT_THE_CALL

REJECT_THE_CALL은 현재의 콜에 대한 거절을 의미한다. 이 경우 콜리가 요청한 콜에 대해 콜리측에서 콜을 거부하도록 스크립트를 등록해 놓은 경우인데, 이 경우, Proxy 서버는 603 (Decline)을 콜리에게 전달한다[4].

LOOP_DETECTED

LOOP_DETECTED는 콜 전환이 서로 루프를 형성하여, 콜을 전달 할 수 없는 경우에 발생한다. 가령 UA1이 UA2에게 호 연결을 시도하려 할 때, UA2는 자신에게 오는 호에 대해 UA3로 전환을 요청했을 때, CPL 서버는 UA3가 등록된 스크립트가 있는지 검사를 한다. 그런데, UA3는 자신에게 오는 콜을 UA2로 전달하라는 요청을 로케이션 서버에 등록했다면, 이 경우 CPL 서버는 루프임을 감지하여, 프락시에게 LOOP_DETECTED를 전달하게 된다. 이를 받은 프락시 서버는 482(Loop Detected) 메시지를 콜리에게 전달한다.

3.2 RequestResponseAction()

프락시 서버는 받은 메시지에 대해, Request인지, Response인지 판단할 수 있다. 받은 메시지가 Response인 경우 RequestResponseAction()을 호출하게 되는데, 동작은 RequestAction()과 거의 유사하다. 다만, 현재 자신이 관리하는 트랜잭션에 있는 메시지인지를 확인한다는 사실과, 함수의 파라미터가 세개로 구성되어 있는데, 앞의 두개는 RequestAction과 동일하다. 세번째 파라미터는 CPL

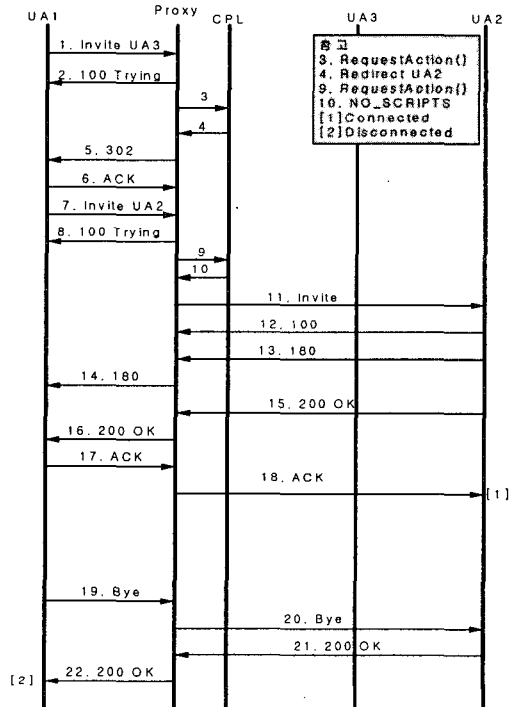
서버로 하여금 현재의 메시지가, 어떠한 형태의 응답인지 판단하는데 도움을 주는 정보이다. 가령 보낸 메시지가 타임아웃 타임동안 무응답상태이면, CPL_PROXY_NOANSWER를 세번째 인자로 주고, 콜리가 busy 상태이면 CPL_PROXY_BUSY를 넘겨준다. 이는 CPL서버가 트랜잭션을 관리하고, 타이머를 관리하는 부담을 줄일 뿐만 아니라, 프락시와 중복된 작업을 수행하지 않도록 구성한 결과이다.

3.3 CPL에서 지원하는 부가 서비스

CPL을 통해 호 전환을 이해하기 위해서는 메시지의 흐름이 필수적이라 할 수 있다. 이러한 부가 서비스의 주요한 예를 살펴보면 다음과 같다.

All Call Forwarding

All Call Forwarding은 자신에게 오는 모든 콜을 제 3자에게 전달하게 하는 호 전환의 기본이다. 예를 들어 UA1 → Proxy → UA2 의 호 설정을 원하는 경우인데, UA2가 자신에게 오는 모든 콜을 UA3로 전달하라고 등록한 경우이다. 이 경우 메시지 흐름이 [그림 4]에 나타나 있다.

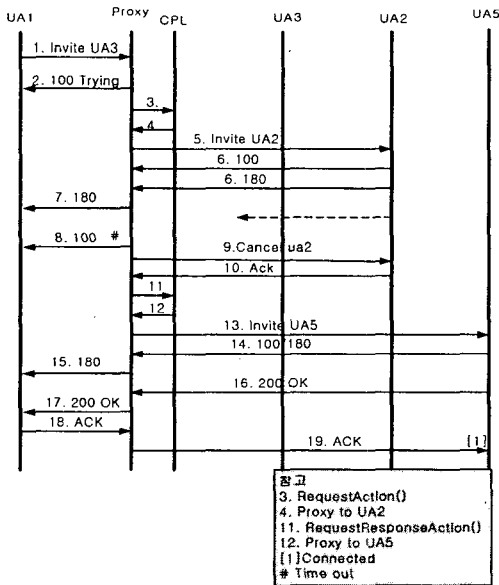


[그림 4] All Call Forwarding의 메시지 흐름

Busy Or No Answer Call Forwarding

Busy Or No Answer Call은 호 전환한 또 다른 콜리가 이미 통화중이거나, 타임아웃이 걸릴 때 까지 커넥션을 맺지 못하는 경우에 대한 부가 서비스이다. 후자의 경우 기존 프락시에서는 콜리에게 CANCEL을 보내고, 콜리에게는 408(Time out)을 전달하게 되는데, CPL을 지원하는 프락시는 408 메시지를 전달하기 전에 CPL 서버에게 또 다른 스크립트가 있는지 요청을 하게 된다.

이때 CPL 서버가 NO_SCRIPTS나 NOTHING_SPECIAL을 주는 경우 프락시 서버는 408 메시지를 전달하지만, PROXY_THE_CALL이나, REDIRECT_THE_CALL을 전달해주는 경우, InterpreterResult_t 구조체의 Location Set으로 프락시 또는 리다이렉트를 하게 된다. 물론, Location Set은 단일 로케이션일 수도 있지만, 리스트 일 수도 있다. 리스트인 경우, 브랜치를 생성하여 포킹을 수행하여, 응답으로 되돌아온 메시지 중 최상의 응답(Best Response)를 찾아 CPL 서버에게 다음 액션을 요청한다. 이러한 동작 플로우가 그림 5에 나타나 있다.



[그림 5] Busy Or No Answer Forwarding의 메시지 흐름

4. 결론 및 향후 연구 과제

본 논문에서는 인터넷 텔레포니 서비스를 이용하기 위해

사용되는 핵심 프로토콜인 SIP 프로토콜을 이용하여 CPL을 지원하여 PSTN에서와 같은 다양한 부가 서비스를 제공할 수 있는 프락시 서버와, CPL 서버의 동작 매카니즘에 대해 살펴보았다.

본 논문에서 구현된 매카니즘은 SIP bis 03을 지원하고, SIP 09의 필수 항목을 지원하며, CPL 표준 스펙을 최대한 반영하였다.

향후 과제로는 현재는 단일 프락시 서버만으로 테스트하였는데, 기존 프락시와, stateless 프락시를 여러 개 두는 토폴로지 상에서 CPL을 이용한 다양한 부가 서비스 테스트 작업 중이다.

또한 리얼 네트워크에 적용하여 성능 평가에 대한 연구가 필요하다.

참고 문헌

[1] 민경주, 이종화, 강신각, “호처리언어(CPL) 지원을 위한 SIP Registrar의 설계 및 구현”, 한국통신학회 2002 하계 학술 발표회 논문집
 [2] RFC 2543-bis 03, draft-ietf-sip-rfc2543bis03.txt
 [3] RFC 2543-bis 09, draft-ietf-sip-rfc2543bis09.txt
 [4] CPL: A Language for User Control of Internet Telephony Services, draft-ietf-iptel-cpl-06.ps
 [5] IETF IPTEL WG,
<http://www.ietf.org/html.charters/iptelcharter.html>