

# 향상된 부가 서비스 지원을 위한 SIP 프락시 서버의 확장 설계 및 구현

민경주 · 이종화 · 강신각 · 박기식

\*한국전자통신연구원

## Extended Design And Implementation of SIP Proxy Server for Improved Additional Internet Telephony Service

Kyoung-Ju Min · Jong-Hwa Yi · Shin-Gak Kang · Ki-Sik Park

\*Electronics and Telecommunications Research Institute

E-mail : {meanrace, jhyi, sgkang, kipark}@etri.re.kr

### 요 약

호 처리 언어(CPL : Call Processing Language)는 공중 전화망에서 제공하는 다양한 부가 서비스를 인터넷 상에서 개발할 수 있는 기술로서, IETF IPTEL 워킹 그룹에서 개발한 표준이다. 사용자들은 CPL을 이용하여 호 전환, 호 방지와 같은 다양한 부가서비스를 요청할 수 있으며, 이것들은 스크립트 형태로 SIP(Session Initiation Protocol) Registrar에 등록되어 실질적인 서비스를 제공받게 된다. 본 논문에서는CPL을 지원하는 프락시 서버의 동작 중, 향상된 부가서비스 지원을 위한 프락시 서버와 CPL 서버간 동작에 대한 설계 및 구현에 대해서 기술하고자 한다. 이러한 향상된 부가 서비스를 지원함으로써, 인터넷 텔레포니(VoIP :Voice over Internet Protocol) 서비스를 이용하는 사용자의 요구 사항 반영에 크게 기여할 수 있다. CPL을 지원하는 SIP 프락시 서버는 리눅스 커널 버전 2.4.x 상에서 C언어를 이용하여 설계 및 구현하였고, CPL 서버는 라이브러리로, 프락시 서버와 연동하여 동작한다.

### ABSTRACT

CPL is a technique that serves various additional service in Internet telephony such as call forwarding, call blocking etc. IETF IPTEL working group developed this CPL standard. Users could request various additional services such as call forwarding, call blocking etc. by registering XML scripts to location servers. This paper would describe the design and the implementation skill of SIP proxy server that support these improved functionalities in detail. SIP registrar and SIP proxy server are designed and implemented in Linux platform because this platform serves fast and low cost development environment.

### 키워드

SIP, CPL, VoIP, Proxy, Registrar, Call Forwarding

### 1. 서 론

인터넷의 급속한 확산으로 인해 IP 망의 이용은 점차로 증가되어 가고 있는 실정이다. 이러한 IP 망을 이용하여 인터넷 텔레포니에 관련된 기술 개발과 표준화가 급속히 진행되어 왔고, 또 진행되고 있다. 이와 더불어 국내의 VoIP 시장 규모가 빠르게 성장하고 있는

추세이다. VoIP 분야에서 가장 대표적인 응용 서비스는 인터넷 전화라 할 수 있는데, 현재까지는 두 사람 사이의 음성 통화 서비스를 제공하는 수준이다. 이에 사용자들은 기본적인 인터넷 전화 서비스는 물론이고, 기존 PSTN(Public Switched Telephone Networks)에서 제공하는 호 전환, 무응답시 전환, 부재중 안내와 같은 다양한 서비스들을 인터넷에서 제공하도록 개발

하려는 노력이 높아지고 있다. 이러한 부가 서비스를 지원할 수 있는 표준 기술의 하나가 호 처리 언어(CPL)이다. 이러한 CPL은 사용자로 하여금 로케이션 서버에 스크립트를 등록하여 서비스를 받을 수 있다 [1,2,7]. 이러한 CPL을 이용하여, 다양한 부가 서비스를 제공해야 한다는 필요성을 인식하지만, 국내에서는 이와 관련된 개발이 부진한 실정이다. 이러한 이유로 본 논문에서는 SIP와 호환을 이루고, 서비스 향상을 위해 CPL을 수용할 수 있는 프락시 서버를 설계하고 구현하였다. 이러한 설계 및 구현은 신속한 개발 환경과 저 비용의 장점을 지닌 리눅스 플랫폼을 기준으로 이루어졌으며, 본 논문의 구성은 다음과 같다.

II 장에서는 CPL을 통해 이루어질 수 있는 부가 서비스에 대해 소개한다. III장에서는 프락시 서버의 설계 및 구현 내용 중 파라미터 핸들링과 관련된 확장 설계 구현에 대해 상세한 사항을 소개한다. 결론 및 향후 과정은 IV 장에서 기술한다.

## II. CPL 기반 부가서비스

사용자는 스크립트를 레지스트라에 저장을 한다. 이때, 사용자는 GUI를 통해 자동으로 스크립트를 생성하는 소프트웨어를 사용할 수도 있고, 메시지 형태로 스크립트를 등록할 수도 있다. 이러한 스크립트는 다양한 사용자의 요구사항을 반영하게 되며, 다음의 그림 1과 같은 전체 흐름도를 통해서 CPL이 지원 가능하며, 이러한 로직을 통해 다음의 표 1과 같은 부가 서비스가 가능하다[1].

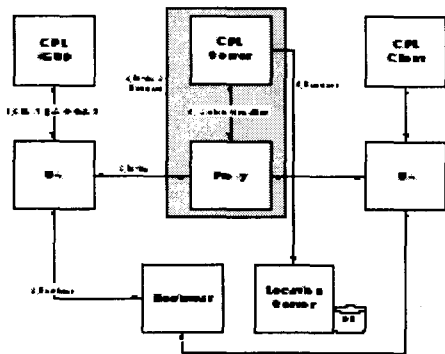


그림 1. CPL 지원을 위한 전체 흐름도

그림 1에서 보는 바와 같이, 사용자는 프락시 서버를 통해서 레지스트라에 등록할 수 있을 뿐만 아니라, 레지스트라에게 직접 메시지를 전달하여, 부가서비스를 등록할 수 있다[1, 3, 4, 5].

표 1. CPL을 통한 지원 가능한 부가서비스

부가 서비스	설명	비고
All Call Forwarding	콜을 지정한 곳으로 전환	
Busy or No Answer	통화 중 또는 무응답시 콜 전환	486 408
Time of Routing	지정된 시간내의 콜 전환	
String Call	제목의 특정 문자열 존재하면 콜 전환	
Call Screening	수신 콜의 특정 사용자, 호스트 거절	
Call Blocking	발신 콜의 특정 사용자, 호스트 거절	
조합된 서비스	위의 서비스 조합된 서비스	

## III. 프락시 서버의 설계 및 구현

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 구조이다. 본 논문에서 다루는 CPL 구조는 첫 번째 형태의 구조를 따르도록 설계 되었다.

### 1. RequestAction, RequestResponseAction

프락시 서버는 수신한 호 설정 메시지에 대해 CPL 서버에게 자신의 행동을 요청하게 된다. 이러한 요청을 수행하는 함수가 RequestAction()인데, 이 함수를 호출하고, 이 함수의 결과에 따라 프락시 서버는 행동을 달리하는데, 동작의 분류를 보면 그림 2와 같다[6].

```

if ((result = RequestAction(pSIPMsg, &pIR)) > 0) {
    switch(pIR->CPLActionType) {
        case NO_SCRIPTS:
        case NOTHING_SPECIAL:
        case PROXY_THE_CALL:
        case REDIRECT_THE_CALL:
        case REJECT_THE_CALL:
        case SEND_CPL_LOG:
        case WRITE_CPL_LOG:
        case LOOP_DETECTED:
    }
}
    
```

그림 2. RequestAction()에 따른 동작 분류

이러한 각각의 행동에 대한 응답이 오면 프락시 서버는 RequestResponseAction()을 호출하여 자신의 다음 행동을 요청하게 되는데, RequestAction()이 SIPMsg\_t, InterpreterResult\_t를 파라미터로 사용하

는데, 비해, Request ResponseAction()은 위의 두 파라미터 이외에 Status 상태를 더 넘겨주는 차이를 제외하면, 두 함수의 기능은 거의 유사하다.

그림 2에서 각각 InterpreterResult\_t를 파라미터로 사용하는데, 이 파라미터는 프락시가 메모리를 할당하고, 주소로 파라미터를 넘겨주게 되면, CPL 서버는 이 구조체에 값을 넣어 프락시에게 보내준다. 관련된 연산 수행후 프락시는 이 구조체의 메모리를 해제 시켜준다. 이러한 InterpreterResult\_t는 CPL이 제공할 수 있는 모든 기능들에 대해 포함하고 있는데, 이 구조체의 구조는 다음의 그림 3과 같다.

```
typedef struct {
    ActionType_t CPLActionType;
    char ActionDesc[INT_RESULT_DESC LENG];
    int ErrorCode;

    union
    {
        InterpretResultProxy_t *proxy;
        InterpretResultRedirect_t *redirect;
        InterpretResultReject_t* reject;
    }Param;

    LocationSet_t *LocationSet;
}InterpreterResult t;
```

그림 3. InterpreterResult\_t의 구조

공용체로 proxy, redirect, reject에 대한 구조를 제한하는데, 각각에 대한 세부 구조가 아래의 그림 4와 같다.

```
typedef struct {
    int timeout;
    unsigned char recurse;
    ProxyOrdering_t ordering;
    int flag;
}InterpreteResultProxy_t;

typedef struct {
    unsigned char permanent;
    int flag;
}InterpreteResultRedirect_t;

typedef struct {
    int flag;
    char *status;
    char *reason;
}InterpreteResultReject t;
```

그림 4. (InterpreterResult\_t \*)->Param의 구조

각각의 파라미터는 호 처리 언어에서 호 전화, 호 설정 방식 등과 같은 기본 동작이 아닌, 내부적인 정책을 결정하는 요인이 된다. 각각의 파라미터에 따른 세부적인 처리는 향상된 부가 서비스 지원을 위해 필수

적이다. 이러한 세부적인 처리에 대해 살펴 보면 다음과 같다.

## 2. Proxy 파라미터의 처리

CPL 서버가 프락시 서버에게 PROXY\_ THE\_ CALL을 넘겨 주었을 때, 프락시 서버는 그림 3의 InterpreterResult\_t의 LocationSet\_t 뿐만 아니라, 그림 4의 InterpreterResultProxy\_t의 구조체를 참조하게 된다. 이때 프락시가 크게 참조할 변수는 timeout, recurse, ordering의 세 가지 정보이다[6].

### ■ timeout 파라미터

timeout은 새로운 로케이션으로 호 전환을 하는데, 시도해 보는 시간을 CPL 서버가 제한하기 위한 파라미터이다. 호 전환의 특성상 기존 프락시에 비해, 많은 시간을 요구하는 경우가 많기 때문에, 이 시간을 CPL 서버가 제한 할 수 있다. 이 파라미터는 100(Trying)이나 180(Ringing) 응답 메시지를 받은 경우, 프락시 서버에서 타임아웃을 일반적으로 32초로 하여 호 설정을 시도하는데, 이 시간을 timeout 파라미터 값으로 변경한다. 이때 CPL 서버가 넘겨주는 이 값이 너무 작거나, 너무 큰 경우, 프락시 서버는 미리 정의된 MAX 또는 MIN 값으로 강제로 설정할 수 있는데, 이는 프락시 서버의 정책에 의해 결정된다.

### ■ recurse 파라미터

recurse는 3xx에 대한 메시지에 대한 응답을 받았을 때, 프락시 서버가 CPL 서버에게 Request ResponseAction() 함수를 호출할 것인지, 호출하지 않고, 기존 프락시의 행동을 취할 것인지에 대한 행동을 제약하기 위한 파라미터이다. recurse가 YES(1)인 경우, 프락시 서버는 CPL 서버에게 RequestResponseAction()을 호출하지만, NO(0)인 경우, CPL 서버를 호출하지 않고, 받은 메시지를 프락싱해 준다.

### ■ ordering 파라미터

ordering은 LocationSet\_t가 리스트 형태로 구성되어 있는데, 포워딩 해야 할 주소가 리스트 형태로 되었을 경우, 리스트에 대해 어떠한 형태로 메시지를 전달할 것인지를 결정한다. 이 때, 이 파라미터는 다시 세부적으로, PO\_PARALLEL, PO\_SEQUENTIAL, PO\_FIRST\_ONLY의 세 값 중 하나를 가질 수 있는데, 디폴트로 PO\_PARALLEL을 갖는다. PO\_PARALLEL의 경우, 기존 프락시가 했던 것과 마찬가지로, Location

Set으로 브랜치를 생성하여 병렬적으로 포킹을 수행한다. PO\_SEQUENTIAL은 프락시가 브랜치에 대해 우선순위가 높은 순서를 정하여, 우선 순위가 높은 것부터 차례로 호 연결을 시도한다. 이때 우선순위는 (LocationSet\_t \*)Location Set ->loc.priorty 값에 의해 결정되며, 만일 우선 순위가 같다면 리스트의 앞에 있는 것을 먼저 보내어 호 연결을 시도한다. 마지막으로 PO\_FIRST\_ONLY는 우선 순위가 가장 높은 것 하나만을 선택하여 호 연결을 시도한다.

### 3. Redirect 파라미터의 처리

CPL 서버가 프락시 서버에게 REDIRECT\_THE\_CALL을 넘겨 주었을 때, 프락시 서버는 그림 3의 LocationSet\_t 뿐만 아니라, 그림 4의 InterpretResultRedirect\_t 구조체를 참조하게 된다. 이때 프락시가 참조할 변수는 permanent 한 가지 정보이다. permanent 파라미터는 YES(1)와 NO(0)의 두 값 중 하나를 갖는다. 이 값에 따라 콜러에게 전달할 메시지가 결정된다. 이 값이 YES 인 경우, 프락시 서버는 301 (Moved Permanent) 응답 메시지를 전달하고, NO 인 경우, 302 (Moved Temporarily) 응답 메시지를 전달한다[6,7].

### 4. Reject 파라미터의 처리

CPL 서버가 프락시 서버에게 REJECT\_THE\_CALL을 넘겨주었을 때 프락시 서버는 Location Set\_t 뿐만 아니라, 그림 4의 InterpretResult Reject\_t의 구조체를 참조하게 된다. 이 때, 프락시 서버는 status와 reason 두 가지 정보를 참조하는데, status는 reject, notfound, busy, failure의 네 개의 스트링 값 중 하나를 갖는다. 이러한 경우, 프락시 서버는 각 각에 대해, 603(Decline), 404(Not Found), 486(Busy Here), 500 (Internal Server Error) 메시지를 만드는데, StartLine의 Reason 필드는 InterpreterResult Reject\_t의 reason 값을 준다. 만일 이 값이 없으며, 디폴트 값으로, 각 각 Decline, Not Found, Busy Here, Internal Server Error를 사용하여 콜러에게 전달한다[6,7].

### 5. 콜 플로우

사용자 UA1이 UA3에게 전화 연결을 시도하는 경우를 예로 들어 본다. 이 때 UA3는 UA2에게 호 전환하는 스크립트를 등록한 상황이다. 이때 <redirect permanent="no"/>와 같은 스크립트를 추가하여 리다이렉트를 할 때 permanent값을 no로 설정하였다. 이러한 경우, 302 (Moved Temporarily)를 콜러에게 전달하고, 그 후의 행동을 취하는 과정이 그림 5에 나타나 있다.

## IV. 결론 및 향후 연구과제

본 논문에서는 RFC2543bis03의 표준을 모두 수용하고, RFC2543bis09의 필수항목을 지원하는 SIP 프락시 서버에 CPL을 지원하여 PSTN에서와 같은 다양한 부가 서비스를 제공할 수 있는 프락시 서버의 설계 및 구현에 관한 상세 사항을 살펴 보았다. 본 논문에서 구현한 SIP 프락시 서버는 CPL 표준 스펙을 최대한 반영하여, 인터넷 텔레포니에서 다양한 부가 서비스 지원을 위한 시스템을 구축하여, 인터넷 전화상에서 PSTN의 부가 서비스뿐만 아니라, PSTN에서 지원할 수 없는 서비스들을 지원할 수 있게 되었다.

향후 과제로는 본 논문에서 구현한 내용이 RFC2543bis09의 필수 항목만을 포함하고 있어, 최근의 표준 동향을 완전히 소화하지 못하고 있다. 따라서, 본 논문에서 구현한 프락시 서버를 RFC2543bis09뿐만 아니라, RFC3261의 표준 스펙을 모두 지원하는 CPL 프락시 서버로의 업그레이드가 필수적이다.

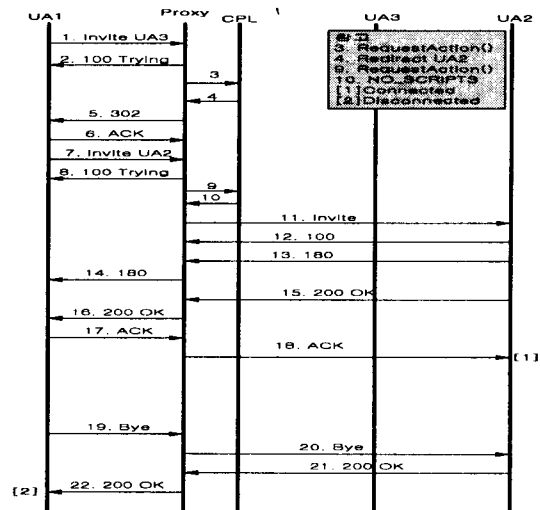


그림 5. All Call Forwarding의 예

### 참고문헌

- [1] 민경주, 이종화, 강신각, “호처리언어(CPL) 지원을 위한 SIP Registrar의 설계 및 구현”, 한국통신학회 2002 하계 학술 발표회 논문 초록집 Vol.25 p.283
- [2] 안상현, 이종화, “VoIP를 위한 호 처리 언어”, 한국정보처리학회지 제 8권 제 2호, 03 2001
- [3] RFC2543bis03, draft-ietf-sip-rfc2543bis03.txt
- [4] RFC2543bis09, draft-ietf-sip-rfc2543bis09.txt
- [5] RFC3261, rfc3261.txt
- [6] CPL: A Language for User Control of Internet Telephony Services, draft-ietf-iptel- cpl-06.ps
- [7] IETF IPTEL WG,  
<http://www.ietf.org/html.charters/iptelcharter.html>, Sep. 20, 2002