

VoIP를 위한 CPL 클라이언트 설계 및 구현

정 옥 조[†] · 이 일 진^{††} · 강 신 각^{†††}

요 약

인터넷상에서 음성을 전달하는 VoIP는 일반 고객들뿐만 아니라 기업에서도 기존 PSTN을 대체하는 수단으로써 각광 받고 있다. 현재의 VoIP는 음성 통화를 효율적으로 제공하고 있으나 VoIP의 촉진을 위해서는 다양한 부가서비스의 지원이 필요하다. 이를 위해 IETF에서 부가서비스 지원을 위한 CPL(Call Processing Language) 표준을 개발하고 있다. CPL을 사용하기 위해서는 사용자가 CPL 스크립트를 작성한 후 특정한 서버에 스크립트를 저장해야 하며 이를 위해 CPL을 지원하는 클라이언트가 요구된다. 본 논문은 VoIP에서 부가서비스를 지원하기 위한 SIP 기반의 CPL 클라이언트 설계 및 구현에 관하여 기술한다. CPL을 지원하기 위한 클라이언트는 리눅스 2.4.x, C, 그리고 GTK1.2를 이용하여 구현하였다.

Design and Implementation of CPL Client for VoIP

Okjo Jeong[†] · Il-Jin Lee^{††} · Shin-Gak Kang^{†††}

ABSTRACT

VoIP that conveys voice in internet is getting into the spotlight as means to alternate existing PSTN in corporation as well as users. Current VoIP is furnishing voice efficiently, but it needs to support various services for VoIP acceleration. IETF is developing CPL standard which is call processing language for supporting various services. User has to store script to specific server for the use of CPL, therefore it is required client to support CPL. This paper describes about design and implementation of SIP-based CPL client for various services. The CPL client was implemented using LINUX 2.4.x, C, and GTK1.2.

키워드 : 인터넷 전화(VoIP), 호 처리 언어(CPL), 스크립트(Script), 클라이언트(Client)

1. 서 론

VoIP(Voice over IP)는 저렴한 비용으로 인해 많은 사용자들을 확보하고 있는 추세이며, 최근에는 일반 사용자뿐만 아니라 기업에서도 VoIP를 적극적으로 활용하고 있다. VoIP는 단말 간의 음성 통화를 효율적으로 제공하고 있으나 VoIP가 활성화되기 위해서는 다양한 서비스들이 요구된다. 부가서비스의 종류로는 호 전환, 무응답 또는 통화 중 착신 전환, 호 예약, 통화 중 대기, 호 필터링과 같은 여러 유형의 서비스가 존재할 수 있다. 고객이 PSTN(Public Switched Telephone Network)에서 이러한 부가서비스를 받고자 하는 경우에 전화 서비스 제공업체에게 해당 부가서비스를 신청하고, 해당 업체는 자신이 보유한 시스템에 서비스를 등록함으로써 서비스를 제공하였다. 그렇지만 이 방법은 서비스 업체가 정의하는 부가서비스만이 사용 가능하며, 따라서 사용자들의 요구에 부합하는 다양한 서비스를 제공할 수 없게 된다. 사용자들이

임의의 시간에 사용이 용이한 방법으로 자신이 직접 원하는 서비스를 등록하여 사용할 수 있도록 제공하기 위하여 사용자 자신 또는 제 3자가 부가서비스를 기술할 수 있는 방법이 필요하다. 이를 위해 IETF Iptel 워킹그룹에서 호 처리 언어인 CPL[1-4]의 표준화를 진행 중이며, CPL을 지원하기 위한 하위 통신 프로토콜로써는 VoIP에서 사용되고 있는 SIP(Session Initiation Protocol)[5-6]와 H.323[7]을 고려하고 있다.

본 논문은 SIP 기반의 통신 환경 하에서 사용자들이 다양한 부가서비스를 생성할 수 있도록 CPL 클라이언트를 설계하였다. 구현된 CPL 클라이언트는 음성 통화 서비스를 제공할 뿐만 아니라 부가서비스를 용이하게 작성할 수 있도록 GUI(Graphic User Interface) 환경을 제공하여 간단한 입력만으로도 스크립트를 생성할 수 있는 기능을 제공한다. 또한 단말이 호를 수신하는 경우 이 호에 대한 부가서비스의 지원 기능도 구현하였다.

본 논문에서는 서론에 이어 2장에서는 SIP 프로토콜 및 CPL 동작 방법에 대해 알아보고, 3장에서는 호 처리 클라이언트 시스템 설계 및 구현에 대하여 설명하고 4장에서는 결

† 정 회 원 : 한국전자통신연구원 통신프로토콜 표준연구팀 연구원
 †† 정 회 원 : 한국전자통신연구원 통신프로토콜 표준연구팀 선임연구원
 ††† 종신회원 : 인터넷 텔레포니 포럼 부의장/운영위원장
 논문접수 : 2002년 11월 6일, 심사완료 : 2003년 7월 16일

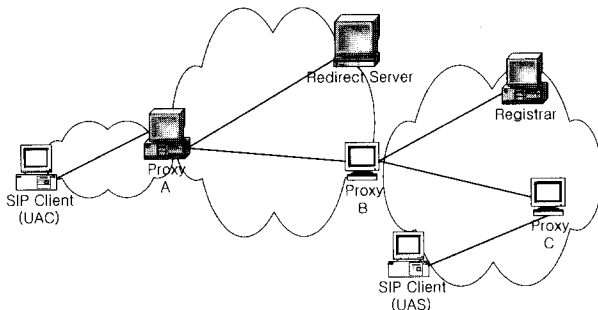
문을 맺도록 한다.

2. SIP 프로토콜 및 CPL 동작 방법

2.1 SIP 프로토콜의 특성

SIP[5]은 HTTP(Hyper Text Transport Protocol)와 유사한 텍스트 기반의 프로토콜로서 사용자 간에 통신을 위한 세션을 시작하는 프로토콜이다. SIP는 2002년 6월에 RFC3261로 확정되었으며 VoIP 서비스의 구현에 적용되고 있다. SIP 메시지는 크게 요청 메시지와 응답 메시지로 구별되며 요청 메시지는 클라이언트에서 서버에게 보내는 메시지로써 INVITE, ACK, CANCEL, BYE, REGISTER, OPTIONS 등이 있다. 응답 메시지는 서버에서 클라이언트로 보내는 메시지로써 각각은 상태 코드로 구분되며, 1xx(Informational), 2xx(Success), 3xx(Redirection), 4xx(Client Error), 5xx(Server Error), 6xx(Global Error)가 있다.

SIP 기반 서비스를 제공하고자 할 때 포함될 수 있는 SIP 구성 요소로는 SIP 메시지 전송을 요구하고 전송된 SIP 메시지를 수신하는 종단 시스템에 해당하는 User Agent, 종단 시스템에서의 다양한 SIP 메시지 전송 요구에 대해 이를 전달해주는 프록시 서버, 사용자의 SIP 메시지 전송 요구에 대해 현재 위치를 알려주어 User Agent가 재 접촉을 하게 하는 리다이렉트 서버, 사용자의 현재 위치를 등록할 수 있게 하며 이를 관리하는 레지스트라 서버로 구성된다. 또한, 레지스트라에 등록된 사용자의 SIP 주소에 대해 Contact 주소를 제공하는 로케이션 서버가 있다. (그림 1)은 레지스트라 서버를 포함한 프록시 서버, 레지스트라 서버를 포함한 SIP 아키텍처를 나타낸다.



(그림 1) SIP 아키텍처

User Agent는 UAC(User Agent Client)와 UAS(User Agent Server)로 구성된다. UAC는 SIP 요청 메시지를 발생시키는 엔티티이며 UAS는 SIP 요청 메시지를 수신하여 응답 메시지를 전송하는 기능을 갖는 엔티티이다. 프록시 서버는 User Agent간 설정되는 호의 상태 정보 유지 여부에 따라 Stateful과 Stateless로 구분된다. Stateful 프록시 서버는 해당 서버에서 서비스하는 호의 상태 정보를 유지함에 따라

Stateless 프록시 서버에 비해 좀 더 다양한 부가서비스를 제공할 수 있는 기능을 지니는 장점을 가진 반면 모든 호에 대하여 상태를 확인해야 하므로 Stateless 프록시 서버에 비해 성능 및 속도 면에 떨어진다. 그러므로 속도가 필요한 백본 망에서는 Stateless 프록시 서버가 사용되며, 사용자에게 다양한 부가서비스를 제공해야 하는 경우에는 Stateful 프록시 서버가 사용된다. 리다이렉트 서버는 수신된 INVITE 메시지에 대하여 해당 주소를 로케이션 서버에서 검색한 후 이 정보를 반환하는 서비스이다. 리다이렉트 서버는 응답 메시지로 3xx 메시지를 사용하며 이 메시지의 헤더에 contact 주소 필드에 다음에 찾아가야 할 주소를 기재한다. 레지스트라 서버는 사용자의 위치정보와 부가정보를 관리하는 기능을 수행한다. 사용자는 레지스트라 서버에게 자신의 위치 정보를 저장하기 위해 유효시간과 함께 주소를 저장한다.

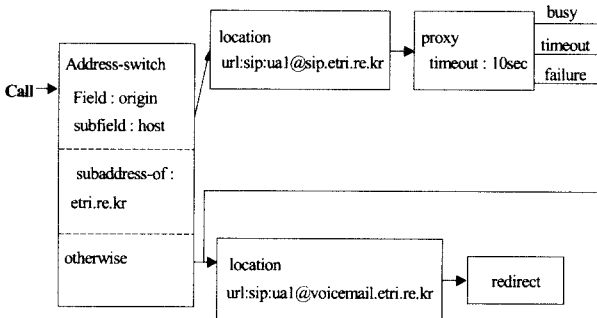
2.2 CPL의 특성

사용자들은 자신이 다른 위치로 이동시에도 전화를 받기를 원하며 또한 특정한 사람이나 특정한 조건에 따라 호를 거절하고 싶어 하기도 한다. IETF에서는 이를 위해 사용자들이 서비스를 정의할 수 있도록 호 처리 언어인 CPL 표준을 진행 중이다. CPL에 대한 표준은 IETF IPTEL 워킹그룹이 구성된 1997년 11월부터 진행되었다. RFC2824[1]는 CPL을 지원하는 프레임 워크와 호 처리를 위한 서비스 모듈에 대한 요구 사항을 기술하는 표준문서로서 2000년 5월에 승인되었다. 현재 작업 중인 드래프트 문서[2]는 CPL 문법을 정의하고 있으며 CPL 스크립트 구조, 신택스, CPL 제공하는 기능, CPL 기반 서비스 예제로 구성되어 있다. CPL은 크게 <incoming>, <outgoing> 그리고 <subaction>으로 구성되며 <incoming>은 수신된 호에 대해 처리 방법을 기술하는 부분이며 <outgoing>은 송신할 때 호에 대해 동작을 지시하기 위해 기술하는 부분이다. 모든 호에 대해 모듈화와 재사용을 위해 서브루틴을 정의할 때는 <subaction> 태그를 사용한다. CPL이 제공하는 기능은 세부적으로 Switches, Location Modifiers, Signalling Operations 그리고 Non-signalling Operations가 있다.

CPL은 XML(Extensible Markup Language) 기반의 언어로서 텍스트 기반 태그로 구성되며 XML 문서들이 갖는 손쉬운 편집 기능, 여러 프로토콜을 이용한 전달 기능, 확장이 쉬운 장점을 가진다. CPL은 하나의 XML 문서 형태를 취하며, CPL 기능들을 적절히 조합함으로써 다양한 VoIP 서비스를 표현할 수 있다. CPL은 SIP를 이용하는 경우 프록시 서버에서 수행된다.

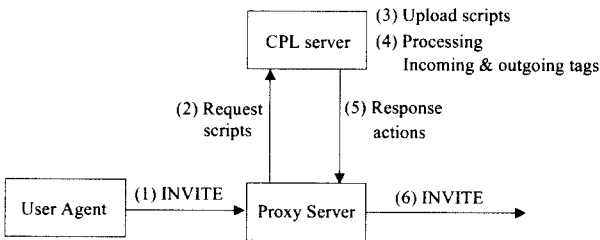
(그림 2)는 CPL 스크립트가 수신된 호에 대해 어떻게 처리하는 지를 보여주는 예를 나타낸다. 만일 CPL을 처리하는 특정 시스템이 호를 수신하는 경우에 호와 관련된 CPL 스크립트를 찾는다. 해당 스크립트가 존재하는 경우에 CPL 스크

립트는 수신된 호의 정보를 사용하여 관련 조건과 일치하는지를 비교한다. 만일 송신자의 host 부분이 etri.re.kr을 포함하고 있다면 CPL 스크립트는 새로운 주소인 sip:ual@sip.etri.re.kr로 호를 proxy하도록 정의한다. 만일 시간 경과 또는 상대방의 통화 중으로 인해 busy 응답이나 실패 응답을 수신할 때는 sip:ual@voicemail.etri.re.kr이라는 음성메일 서버에 호를 전환함으로써 음성메시지를 남길 수 있도록 지정하고 있다.



(그림 2) CPL 스크립트 동작 과정

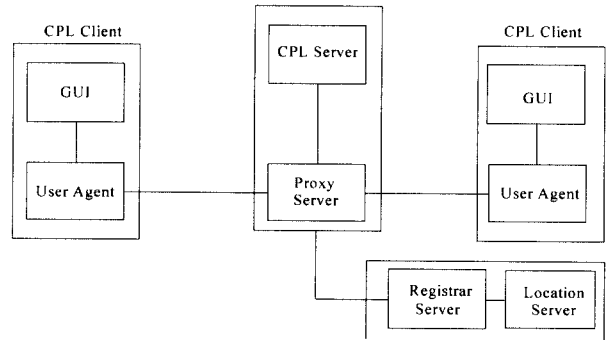
(그림 3)은 CPL 지원을 위한 동작 구조를 나타낸다. 우선 사용자는 GUI나 일반 편집기를 사용하여 CPL 스크립트를 작성하고, 작성된 CPL 스크립트를 특정한 서버에 등록하는데 SIP에서는 레지스트라 서버에 해당한다. 다른 사용자가 그림과 같이 INVITE 메시지를 보내서 호 연결을 시도하는 경우에 SIP 프록시 서버는 수신된 INVITE 메시지를 CPL 스크립트를 처리하는 모듈로 보낸다. 그림에서 CPL 서버라고 명명된 부분이 CPL 스크립트를 처리하는 부분이다. CPL 서버는 해당 호에 대한 CPL 스크립트를 서버로부터 업로드하여 <incoming> 또는 <outgoing> 태그의 조건과 비교한다. 만일 주어진 조건과 일치한다면 그에 따른 동작 방법을 프록시 서버에게 전달하게 되고 프록시 서버는 이를 수행한다. 만일 등록된 CPL 스크립트가 없다면 프록시 서버는 기존의 자신의 동작대로 처리한다.



(그림 3) CPL 지원을 위한 동작 구조

2.3 CPL 지원을 위한 아키텍처

CPL을 지원하기 위한 아키텍처는 SIP을 기반으로 하는 경우에 (그림 4)와 같은 구조를 갖는다.



(그림 4) CPL 지원을 위한 구조

각 구성 요소들로는 CPL Client, 프록시 서버, CPL 서버, 레지스트라 서버, 로케이션 서버로 구성된다. CPL 클라이언트는 CPL을 지원하는 클라이언트를 의미하며 두 계층으로 구별할 수 있다. 상위 계층은 VoIP 서비스를 표현하는 응용 모듈이 존재하고 하위 계층은 SIP 메시지의 생성, 전송, 수신, 처리를 담당하는 User Agent 모듈이 존재한다. 또한 User Agent와 함께 RTP 스택이 존재하며 이는 User Agent를 통해 호가 설정된 이후 호 설정을 통해 협상된 미디어 세션 정보를 이용하여 음성 정보를 전달하는데 활용된다. 프록시 서버는 수신된 메시지에 대하여 상태정보를 유지하면서 호를 네트워크의 다음 노드로 전달하는 stateful 프록시와, 상태 정보를 유지하지 않고 단지 호를 네트워크의 다음 노드로 전달하는 stateless 프록시의 두 가지 모드로 동작할 수 있다. 프록시 서버가 메시지를 수신하면 다음 노드를 알기 위해 로케이션 서버를 통해 정보를 얻는다. CPL 서버는 프록시 서버가 요청한 요청 혹은 응답 메시지에 대한 부가서비스를 검증하기 위해서 로케이션 서버로부터 해당하는 서비스 로직인 CPL 스크립트를 검색하여 가져와서 XML 파서를 통해 파싱한 후 해당 스크립트를 해석한다. 수신된 메시지의 헤더 중 특정한 정보와 일치하는 스크립트의 조건이 존재하면 조건에 따라 수행하도록 지정되어 있는 스크립트의 결과를 생성한다. 이 정보는 프록시 서버에 전달되어 프록시 서버의 행동을 제어하게 된다. 레지스트라 서버는 사용자로부터의 현재 위치 관리를 해주는 SIP 서버로서 SIP 프로토콜의 REGISTER 메시지의 처리를 담당한다. 레지스트라 서버는 사용자의 위치 정보를 관리하기 위하여 데이터베이스를 사용하게 되며 SIP 메시지의 처리를 위한 파싱 엔진을 사용한다. 레지스트라 서버는 로케이션 서버로 사용되는 DBMS에 사용자의 등록 정보를 삽입, 삭제, 검색, 갱신하는 기능을 수행하게 된다. CPL 지원을 위하여 레지스트라 서버는 스크립트에 대한 등록, 갱신, 검색, 삭제 기능을 수행한다.

3. CPL 지원을 위한 클라이언트 설계 및 구현

CPL을 지원하기 위한 클라이언트 시스템은 기본적으로 음성 통화를 지원하도록 구현하였다. 또한 부가서비스를 정의하

는 CPL 스크립트를 손쉽게 생성할 수 있도록 GUI 환경을 제공하여 간단한 입력만으로도 자동적으로 스크립트를 생성할 수 있도록 설계 하였으며, 단말이 호를 수신하는 경우 이 호에 대한 부가서비스의 지원 기능도 설계하였다. CPL 클라이언트는 SIP RFC2543bis03[5]을 기반으로 한 SIP 시스템 상에서 IETF Iptel 워킹 그룹의 RFC2824 및 draft-ietf-iptel-cpl-06 표준문서에 따라 구현하였으며, CPL 스크립트를 전달하기 위한 방법으로 SIP REGISTER 메시지[3]를 사용하였다. 본문에서는 호 설정 및 부가서비스를 위한 CPL 클라이언트의 기능 중에서 일반적인 음성 통화 기능 보다는 CPL 지원 기능에 관해 중점적으로 기술한다.

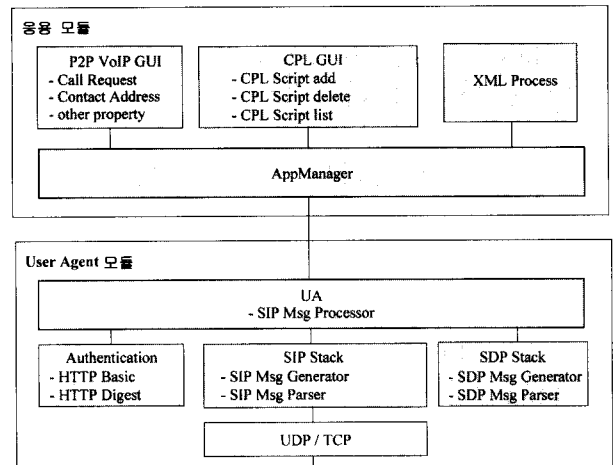
3.1 CPL 스크립트 등록 기능 설계

CPL 기반의 VoIP 부가서비스로는 여러 가지가 있지만 구현은 아래에 기술된 각 서비스들을 제공하도록 CPL 클라이언트를 구현하였으며 이들 서비스들은 크게 3 그룹으로 구성된다.

- 호 전환 서비스(CFS : Call Forwarding Service) : 조건을 명시하고 조건이 만족되는 경우 수신 받는 호를 전환시키는 서비스
 - All Call Forwarding 서비스
 - Busy or No Answer Call 서비스
 - Time Based Call 서비스
 - Address Based Call 서비스
 - String Based Call 서비스
- 호 필터링 서비스(CSS : Call Screening Service) : 송신자가 임의의 수신자에게 전화 통화를 요청하는 경우 통화를 요청할 수 있는지 검증한 후 호를 연결시키는 서비스 유형으로 사업자 또는 관리자가 각 사용자의 통화 범위를 제한시키는 서비스
- 호 거절 서비스(CBS : Call Blocking Service) : 수신된 통화에 대하여 조건을 명시하고 조건이 만족되는 경우 통화 요청을 자동적으로 거절하는 서비스

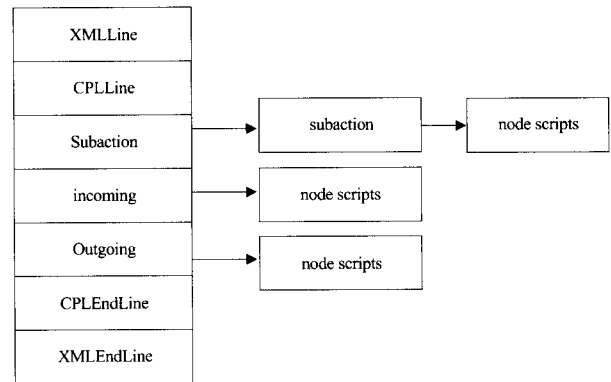
부가서비스를 사용하기 위해서는 우선 사용자, 통신사업자 또는 기타 사업자들이 부가서비스를 정의하고 시스템에 등록하는 과정이 요구된다. 이를 위해 CPL 클라이언트는 (그림 5)와 같은 기능 모듈을 갖는다.

CPL 클라이언트는 음성 통화 기능을 지원하기 위하여 P2P VoIP GUI를 제공한다. 이 GUI를 통해 사용자들은 통화하고자 하는 상대방의 SIP 주소를 입력함으로써 음성통화를 수행할 수 있다. 또한 사용자 위치 정보의 등록을 위한 contact address 등록 기능도 제공한다. CPL GUI는 사용자가 스크립트를 생성할 때 XML과 CPL의 문법 구조를 알지 못하는 경우 간단하고 편리한 입력방식을 제공한다. CPL GUI에서 입력한 값들은 AppManager에서 XML 파서를 이용하여 스크립트



(그림 5) SIP 기반 CPL 수용을 위한 클라이언트 구조

에 대한 유효성 검사를 진행한다. 유효성 검사가 성공적으로 수행되면 AppManager는 자동으로 스크립트를 생성하기 시작한다. (그림 6)은 사용자의 정보를 사용하여 스크립트를 생성하기 위해 사용하는 구조체를 나타낸다. 구조체는 XML 임을 나타내는 필드와 CPL 임을 나타내는 CPL 필드를 가지며 Subaction, incoming, outgoing을 가지고 있다. 각 Subaction, incoming, outgoing 필드들은 하위의 노드를 가질 수 있다.

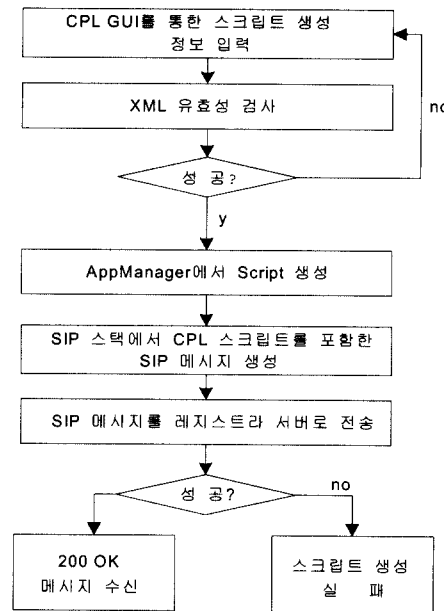


(그림 6) CPL 스크립트 생성을 위한 구조체

User Agent 모듈은 응용 모듈로부터 수신된 스크립트를 레지스트라 서버로 전송하기 위해 SIP 메시지를 생성하는 기능을 수행한다. SIP User Agent는 응용 모듈로부터 생성된 CPL 스크립트를 레지스트라 서버에 등록하기 위한 등록 기능, 등록된 CPL 스크립트를 제거하기 위한 삭제기능, 이미 저장된 CPL 스크립트의 상태 확인을 위한 검색 기능을 제공하도록 설계하였다. CPL 지원을 위해 필요한 헤더는 Content-Disposition 헤더, Accept-Disposition 헤더, If-Unmodified-Since 헤더가 있다. Content-Disposition 헤더는 SIP 스펙에서 제공하고 있지만, CPL 등록 기능을 위해서 필드의 기능 확장이 필요하다.

Content-Disposition 헤더는 REGISTER 요청 메시지와 응답 메시지에서 메시지 바디의 목적을 기술하는데 사용된다. 이 헤더는 disposition, action과 modification-date 파라미터를 가진다. Disposition 타입은 script와 sip-cgi의 두 개를 정의하며 action 파라미터는 스크립트를 업로드시 서버가 스크립트의 저장 기능 또는 삭제 기능을 수행하여야 할지를 나타내는데 사용한다. 만일 0보다 큰 길이를 가진 스크립트가 바디에 포함되면 action="store"가 주어져야 하며 action="remove"는 0 문자 길이를 수반할 때에만 사용된다. modification-date 파라미터는 스크립트가 최근에 변경된 시간을 나타내는데 사용된다. Accept-Disposition 헤더는 타입이 클라이언트나 서버에서 수신 가능한 disposition을 나타내기 위해 사용된다. 메타 타입인 *는 모든 콘텐츠 disposition 타입을 나타내며 어떠한 콘텐츠 disposition 타입도 수신 가능하다는 것을 의미한다. Action과 modification-date 파라미터는 Accept-Disposition에서는 의미를 가지지 않는다. If-Unmodified-Since 헤더는 request를 조건적으로 만드는데 사용된다. 만일 요청된 자원이 이 필드에 명시된 시간 이래로 변경되지 않았다면, 서버는 If-Unmodified-Since 헤더 필드가 없는 것처럼 요청된 동작을 수행하여야 한다. 만일 요청된 자원이 지정된 시간 이래로 변경되었다면, 서버는 요청된 동작을 수행하지 않아야 하고 412 Precondition Failed 메시지를 반환한다. 이 과정을 거치면 User Agent는 REGISTER 메시지를 생성하게 되며 이 메시지를 네트워크로 전송함으로써 레지스트라 서버에 전달하게 된다. CPL 스크립트의 등록 기능은 해당 스크립트를 SIP REGISTER 메시지의 바디 부분에 위치시켜 전송함으로써 수행된다. 메시지의 바디 부분의 내용을 명시적으로 나타내기 위해 Content-Type 헤더 필드에 MIME 타입인 application/cpl+xml을 할당하고 Content-Disposition 헤더 필드에 스크립트를 저장하도록 값을 설정하여 전송한다. 이렇게 전송된 CPL 스크립트는 레지스트라 서버를 통해 등록된다. (그림 7)은 CPL GUI를 사용하여 CPL 스크립트를 생성하고 등록하는 과정을 나타낸다.

사용자가 레지스트라 서버에 이미 저장된 CPL 스크립트를 삭제하고자 할 때는 Content-Disposition 헤더 필드에 해당하는 MIME 타입을 할당하고 Content-Disposition 헤더 필드에 스크립트를 삭제하도록 값을 설정하며 Content-length 헤더 필드를 0으로 설정하여 REGISTER 메시지를 전송함으로써 삭제 기능을 수행한다. 레지스트라 서버는 이러한 헤더 정보를 수신하는 경우에 해당되는 CPL 스크립트를 삭제한다. CPL 스크립트의 검색 기능은 등록과 삭제 과정 중에 반환되는 응답 메시지를 사용하여 검색과정이 수행된다. 레지스트라 서버는 클라이언트가 스크립트에 대해 등록이나 삭제를 요구할 때 그 응답으로 응답 메시지를 전송하는데 이 메시지에 현재 등록되어 있는 스크립트를 함께 보낸다. SIP User Agent는 수신된 응답 메시지를 분석하여 상위의 응용 모듈로 CPL 스크립트를 전달하는 기능을 수행한다.



(그림 7) CPL 스크립트 생성 flowchart

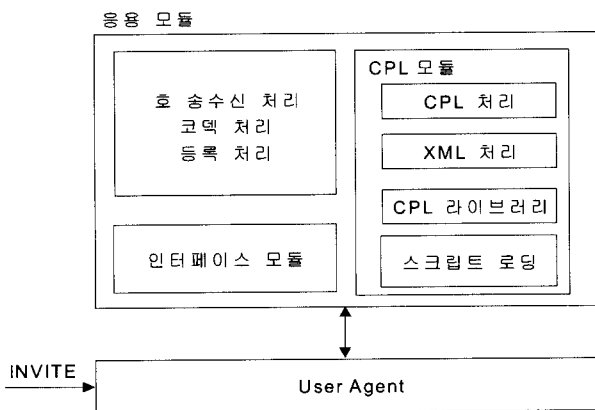
RFC2543bis03에서 정의되어 있는 두 가지 인증 방식(HTTP Basic & Digest)을 이용하여 사용자를 인증할 수 있다. CPL 클라이언트가 레지스트라 서버에 REGISTER 메시지를 전송할 때 레지스트라 서버는 마찬가지로 사용자 인증을 요구하도록 구현하였다. 레지스트라가 401 메시지를 보내오는 경우 User Agent는 응용 모듈에게 전달하여 사용자로부터 추가 인증 정보를 전달 받는다. 이는 사용권한이 없는 사용자가 임의로 스크립트를 레지스트라 서버에 저장하는 것을 방지한다.

SIP 스택 모듈은 User Agent에서 생성한 헤더 정보를 사용하여 SIP 메시지를 생성한다. SIP REGISTER 메시지는 Content-Disposition 헤더, Accept-Disposition 헤더, If-Unmodified-Since 헤더에 값을 할당하여 REGISTER 메시지를 생성한다. 이렇게 생성된 SIP REGISTER 메시지는 UDP 또는 TCP 소켓을 이용하여 네트워크를 통해 레지스트라 서버로 전송된다.

SIP에서 단말 간에 프록시 서버 없이 직접 호를 연결할 수도 있는데 이 경우에 수신하는 단말에 부가서비스 기능을 추가함으로써 단말에서 부가서비스를 지원할 수 있다. 이를 위해 CPL 클라이언트에 부가서비스 처리 기능을 구현하였다. 단말에 부가서비스 기능을 지원하도록 CPL 모듈을 구현하면 직접 호 연결 요청의 경우뿐만 아니라 프록시 서버가 문제가 발생할 경우에도 단말 단에서는 부가서비스 기능을 지원할 수 있게 된다. (그림 8)은 CPL 클라이언트에서 호를 수신할 때 호 처리 기능을 구현하기 위한 모듈을 나타낸다.

CPL 클라이언트가 INVITE 메시지를 수신하면 User Agent는 SIP 파서를 사용하여 메시지를 파싱한 후 상위의 응용 모듈에 파싱된 메시지 정보를 전달한다. 응용 모듈은 이 정보를 사용하여 로케이션 서버에게 관련된 스크립트가 등록

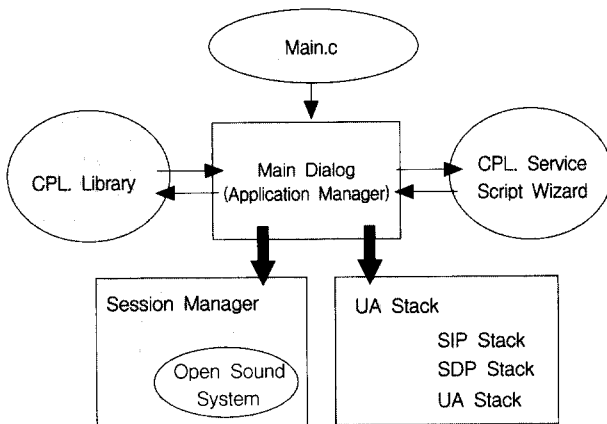
되어 있는지를 질의한다. 만일 등록된 스크립트가 있다면 CPL 모듈이 스크립트를 로딩한 후 XML 파서를 사용하여 파싱한 정보를 DOM(Document Object Model) 형태로 메모리에 유지한다. CPL 처리 모듈은 CPL 라이브러리를 사용하여 현재 수신된 호에 대하여 특정한 절차가 기술되어 있는지 파악한다. CPL 처리 모듈은 proxy, redirect, reject의 세 가지 실행을 지시하는 결과를 발생시키는데 본 구현에 있어서는 구현의 용이성을 위하여 redirect와 reject의 기능만을 현재 지원하도록 구현하였다. CPL 처리 모듈은 스크립트를 해석하여 결과가 나오면 User Agent 모듈에게 그 결과의 실행을 지시함으로써 호 전환 및 호 거절 서비스를 지원하도록 하였다.



(그림 8) 클라이언트의 호 처리 기능

3.2 VoIP를 위한 CPL 클라이언트 구현

CPL 지원을 위한 클라이언트 구현은 리눅스 7.2 레드햇 2.4x 커널상에서 GTK1.2를 사용하여 GUI를 구성하였으며 C 언어를 사용하여 상세 기능을 구현하였다. 그리고 XML 파서는 일반적으로 사용되고 있는 아파치의 Xerces XML 파서를 사용하였다. 또한 RTP 부분은 Linux의 OSS(Open Sound System)를 사용하여 구현하였다. (그림 9)는 CPL 클라이언트 구현 다이어그램을 나타낸다.



(그림 9) CPL 클라이언트 구현 다이어그램

CPL 클라이언트 응용모듈은 <표 1>과 <표 2>에서 나타난 함수들을 사용한다. 응용모듈은 사용자의 위치 정보를 등록하는 등록기능과 음성 통화를 위한 호 설정 및 해지를 제공하는 호설정 해지기능, 상대방의 능력 정보를 획득하는 능력 전달기능, 그리고 GUI의 도움 없이 CPL 스크립트를 직접 작

<표 1> 응용 모듈 관련 함수

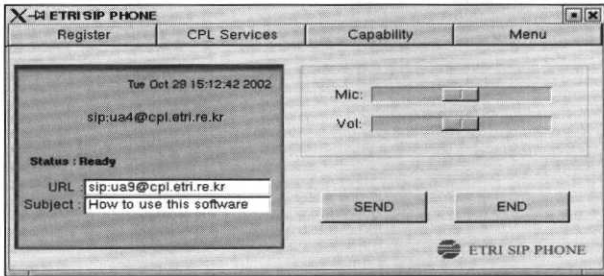
응용 모듈 기능	관 련 함 수
등록 기능	Registration() RegisterCnf()
호 설정/해지기능	EstablishConnection() InformationInd() acceptConnection() RejectConnection() CallSetup Ind() CallSetup Cnf() ReleaseConnection() CallReleaseIndication() NotSupportedMedia Connection() CallCancelIndication() RedirectCallCpl() RejectCallCpl()
능력 전달 기능	GetCapabilities() CapabilitiesCnf() GetCapabilities Retry()
사용자 선택전달기능	UpdateUIInfo()
기타 기능	GetWriteMessage() WriteMessage() ResetAppManager()

<표 2> CPL 관련 함수

gint button_clicked (GtkObject *code)	CPL 관련 버튼이 눌렀을 때 처음 호출되는 함수. Menu, CPL_Add, CPL_List, CPL_Delete 함수를 각각 호출한다.
void ui_Menu()	Menu 버튼을 눌렀을 때 호출되는 함수. MenuWindows() 함수를 불러 메뉴 관련 다이얼로그를 띄우고 윈도우 컨트롤을 한다.
void MenuWindows()	메뉴관련 다이얼로그를 만들고 각종 컨트롤을 붙여 사용자의 입력을 받을 수 있게 한다.
void ui_CPL_Add()	Add 버튼을 눌렀을 때 호출되는 함수. ui_CPL_Add_Active() 를 불러 Add Service 관련 다이얼로그를 띄우고 윈도우를 컨트롤한다.
void ui_CPL_Add_Active()	여러 스크립트를 생성할 수 있는 add 함수. 관련 다이얼로그를 만들고 여러 컨트롤을 붙여 사용자의 입력을 받을 수 있게 한다.
void ui_CPL_List()	List 버튼을 눌렀을 때 불리는 함수. CPLRequestList() 함수를 불러 list 명령을 수행한다.
int CPLRequestList()	CPLRequestList_Thread 함수를 스레드로 띄워 list 명령을 수행한다.
int ui_CPL_Delete()	Delete 버튼을 눌렀을 때 불리는 함수. CPLRequestDelete 함수를 불러 Delete 명령을 수행한다.
int CPLRequestDelete()	CPLRequestDelete_Thread를 띄워 Delete 명령을 수행한다.

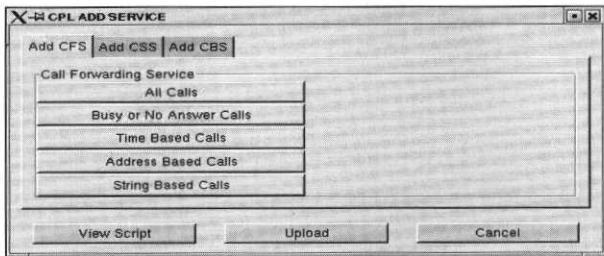
성하여 레지스트라 서버에 등록할 수 있도록 제공하는 기능 등을 제공한다.

일반 사용자들이 스크립트를 손쉽게 작성하기 위해서는 단순하면서도 편리한 사용자 입력 환경을 제공할 필요가 있다. (그림 10)은 리눅스 상에서 구현된 SIP 기반의 인터넷 전화 GUI를 나타내며 상단의 CPL Service 메뉴는 CPL 스크립트를 생성하기 위한 메뉴이다. 음성 통화를 위해서는 우선 Register 메뉴를 사용하여 사용자 정보를 등록한다. 그리고 URL에 통화하고자 하는 상대방의 SIP URL을 입력한 후 SEND를 누름으로써 통화를 시작한다.



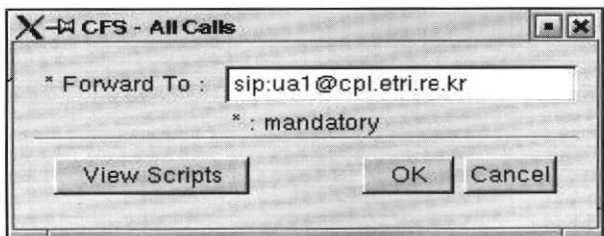
(그림 10) ETRI SIP 전화 GUI

CPL Service 메뉴는 (그림 11)과 같이 구성된다. 이것은 CPL 스크립트 생성을 위한 메인 메뉴를 나타내는 GUI이며 사용자는 CPL 서비스 메뉴에서 원하는 메뉴를 선택하여 해당하는 값을 입력함으로써 CPL 스크립트를 생성할 수 있다.



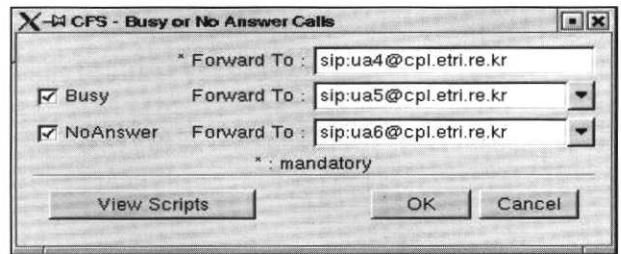
(그림 11) CPL 스크립트 생성을 위한 GUI

(그림 12)는 호 전환 서비스 메뉴 중에서 All Call Forwarding 서비스 메뉴를 나타낸다. 이 서비스는 자신에게 오는 모든 호에 대하여 항상 sip:ua1@cpl.etri.re.kr 주소로 호를 전환한다. 따라서 사용자는 자신이 이동하는 곳에서 항상 호를 수신할 수 있게 된다.

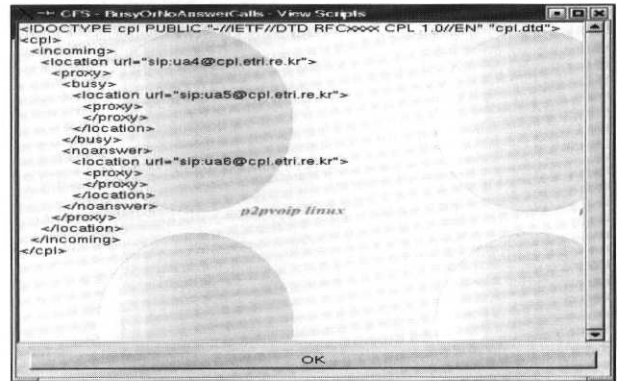


(그림 12) All Call Forwarding

(그림 13)은 호 전환 서비스 메뉴 중 Call forwarding Busy/No Answer 서비스를 위한 메뉴를 나타낸다. 이 서비스는 자신의 전화가 통화 중 혹은 응답이 없을시 상대방 전화를 사용자가 입력된 주소로 호를 전환하는 기능을 하게 된다. (그림 14)은 GUI를 통해 작성한 값들을 사용하여 생성한 CPL 스크립트를 나타낸다.



(그림 13) Busy or No Answer Service



(그림 14) CPL 스크립트 예

Time Based Calls 서비스는 사용자가 지정한 시간을 기준으로, Address Based Calls 서비스는 송신자의 주소를 기준으로, 그리고 String Based Calls 서비스는 SIP 메시지 헤더의 문자열을 기준으로 호를 전환할지를 결정하도록 제공하는 서비스이다. 그리고 Call Screening Service(CSS)는 송신시 조건에 만족하는 경우는 발신호를 제한하는 서비스이며 Call Blocking Service(CBS)는 조건에 만족하는 경우는 자동으로 수신호를 거절하는 서비스를 나타낸다. CSS 서비스는 국제 전화 등을 제한하고자 할 때 사용할 수 있으며, CBS는 광고성 전화를 거절할 때 유용하게 사용될 수 있다.

4. 결 론

인터넷 전화 사업자들은 사용자들의 요구에 따라 다른 곳으로 호를 전환하거나 통화중에 인터넷 음성메일 시스템을 사용한다거나 수신을 거부할 수 있는 다양한 기능들을 제공하기 위해서 사용자들이 부가서비스를 정의할 수 있도록 제공할 필요가 있다. 본문에서 기술한 CPL 클라이언트 시스템은 음성 통화를 기본적으로 지원하도록 구현하였으며 부가서

비스를 정의하는 CPL 스크립트를 손쉽게 생성할 수 있도록 GUI 환경을 제공하여 간단한 입력만으로도 자동적으로 스크립트를 생성하여 등록할 수 있는 기능을 구현하였다. 또한 단말이 호를 수신하는 경우에도 이 호에 대한 부가서비스의 지원 기능도 구현하였다. 이를 통해 인터넷 전화 서비스는 PSTN에서 지원하는 서비스들을 수용할 수 있을 뿐만 아니라 웹, 전자메일 등의 인터넷 서비스들과의 연계를 통해 다양한 부가서비스의 지원도 가능하게 된다. 향후 과제로는 본 논문의 설계 및 구현이 SIP 표준문서인 RFC2543bis03을 기반으로 구현되어 있으므로 최근에 확정된 RFC3261 표준을 추가로 반영해야 하며 호 처리 시스템의 실제적인 테스트의 진행이 요구된다.

참 고 문 헌

- [1] J. Lennox, H. Schulzrinne, "Call Processing Language Framework and Requirements," IETF RFC2824, May, 2000.
- [2] J. Lennox, H. Schulzrinne, "CPL : A Language for User Control of Internet Telephony Services," IETF draft-ietf-iptel-cpl-06, Jul., 2002.
- [3] J. Lennox, H. Schulzrinne, "Transporting User Control Information in SIP REGISTER Payloads," IETF draft-lennox-sip-reg-payload-01, Oct., 2000.
- [4] 이종화, 강신각, "VoIP 호 처리 언어 기술 동향", 한국통신학회 학회지, Vol.19, No.2, 2002.
- [5] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP : Session Initiation Protocol," IETF draft-ietf-sip-rfc2543bis-03, May, 2001.
- [6] J. Rosenberg, H. Schulzrinne, et al, "SIP : Session Initiation Protocol," RFC 3261, Jun., 2002.
- [7] ITU-T Recommendation H.323, "Packet-based multimedia communications systems," 1998.



정 옥 조

e-mail : okjo@etri.re.kr
 1995년 서울시립대학교 전산통계학과(학사)
 2001년 서울시립대학교 전산통계학과(이학 석사)
 2001년~현재 한국전자통신연구원 통신프로토콜 표준연구팀 연구원

관심분야 : 인터넷 멀티캐스트, VoIP, QoS

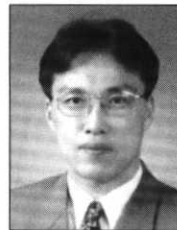


이 일 진

e-mail : il1024@etri.re.kr
 1994년 충북대학교 전자공학과(공학사)
 1996년 충북대학교 전자공학과 통신공학(공학석사)
 1996년~1999년 삼성항공(주) 전임연구원
 1999년~2000년 삼성전기(주) 전임연구원

2000년~현재 한국전자통신연구원 통신프로토콜 표준연구팀 전임연구원

관심분야 : VoIP, 디지털 통신



강 신 각

e-mail : sgkang@etri.re.kr
 1984년 충남대학교 전자공학과(학사)
 1998년 충남대학교 전자공학과(박사)
 1995년 정보통신기술사
 1984년~현재 한국전자통신연구원 통신프로토콜 표준연구팀 팀장/책임연구원

1997년~현재 ITU-T SG 7 Rapporteur

2000년~현재 인터넷 텔레포니 포럼 부의장/운영위원장

관심분야 : 멀티캐스트 통신, VoIP, 인터넷 보안