

# SIP Registrar에서의 SIP URI 비교를 위한 API 설계

박선옥<sup>o</sup> · 현욱 · 강신각

한국전자통신연구원

## Design of API for SIP URI Comparison on SIP Registrar

Sun-Ok Park<sup>o</sup> · Hyun Wook · Shin-Gak Kang

Electronics and Telecommunications Research Institute

E-mail : {sunoko, whyun, sgkang}@etri.re.kr

### 요 약

컴퓨터를 통한 커뮤니케이션의 수단을 보다 인간 중심으로 바꾸고 있는 VoIP 관련 기술들은 최근들어 신속하게 시장의 뜨거운 흐름을 타고 있다. 초창기 인터넷 텔레포니 서비스들은 ITU-T의 H.323을 기반으로 제공되었으나, 여러 가지 문제점들로 인하여 IETF의 SIP를 이용한 서비스 모델로 변화하고 있는 추세이다. SIP 프로토콜은 인터넷 텔레포니 서비스를 제공하기 위해, 통신하고자 하는 상대방의 위치를 찾아 단말간의 호 설정, 호 정보 수정, 호 해지 등의 기능을 제공하는 응용계층의 호 제어 프로토콜이다. SIP 프로토콜에서 개인의 이동성을 보장하기 위해서는 사용자들의 위치가 변경될 때마다 현재 위치를 Registrar에게 등록 시켜야 하며, 이러한 등록 정보들은 이후 Proxy 서버나 Redirect 서버를 위한 Location Service 제공을 위해 사용된다. Registrar에서는 사용자의 위치정보에 대한 등록, 수정, 삭제, 조회에 대한 요청 메시지 수신 시, SIP URI 비교 API가 요구된다. 본 논문에서는 RFC3261 기반의 SIP Registrar의 기본 동작을 간단히 소개하며, SIP URI 비교 시 요구되는 기능들을 분석하고, API를 설계한다.

### 키워드

SIP, Registrar, SIP URI, URI Comparison

## 1. 서 론

국내외 인터넷 분야에서 VoIP에 대한 관심이 증대되면서 IETF에서 ITU-T의 H.323의 대안으로 SIP (Session Initiation Protocol)를 IP 텔레포니 서비스를 위한 호 처리 프로토콜로 제정하였다.

SIP는 IETF 산하 MMUSIC WG에서 처음 개발한 프로토콜로서, 99년 3월 RFC2543으로 확정되었다가, SIP WG이 구성되면서 SIP 프로토콜에 대한 개정작업들이 진행되었다. SIP WG에서 2000년 6월 rfc2543bis draft가 처음으로 제안된 이후 2002년 2월말 bis-09 버전이 제안되었으며, 이를 기반으로 2002년 7월초 RFC3261 표준이 제정되었다. 현재, 대부분의 국내외 SIP 기반 인터넷폰은 bis-03 버전을 기준으로 개발된 상태이며, 2002년 7월초 RFC3261 표준이 제정된 이후 RFC3261로의 재개발을 추진하고 있는 중이다.

SIP 프로토콜은 인터넷 텔레포니 서비스를 제공하

기 위해 통신하고자 하는 상대방의 위치를 찾아 단말간의 호 설정, 호 정보 수정, 호 해지 등의 기능을 제공하는 응용계층의 호 제어 프로토콜이다. SIP 프로토콜에서 개인의 이동성을 보장하기 위해서는 사용자들의 위치가 변경될 때마다 현재 위치를 Registrar에게 등록 시켜야 하며, 이러한 등록 정보들은 이후 Proxy 서버나 Redirect 서버를 위한 Location Service 제공을 위해 사용된다.

Registrar 서버는 사용자의 현재위치를 등록 받는 서버로서 REGISTER 메소드내의 Contact 헤더와 Expire 헤더들의 조합으로 현재 사용자의 위치에 대한 등록/갱신/삭제/조회 등의 기능을 담당한다. 사용자로부터 등록 받은 정보는 Registrar DB에 저장되며, 이후 Proxy 서버나 Redirect 서버 등에게 Location Service를 제공하기 위해 사용된다.

사용자의 AOR(Address-of-Record) URI와 하나 이상의 contact address에 대한 바인딩(binding) 정보

가 DB에 기록되며, 이후 Proxy 서버가 도메인의 Location Service를 제공해야 할 Request 메시지를 수신하게 되면 Request-URI와 일치하는 AOR URI의 contact address로 Request 메시지를 전달한다. [그림 1]은 간단한 contact address 등록 절차 및 Proxy 서버에서의 Location Service 이용 절차를 보여준다.

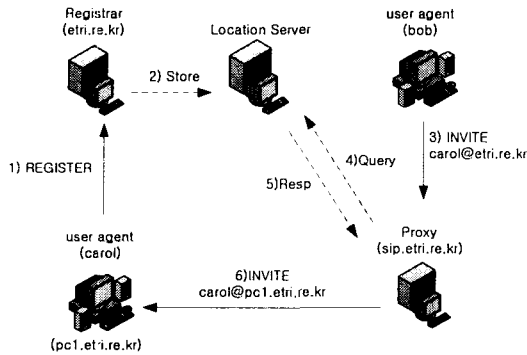


그림 1 REGISTER Example

user agent는 REGISTER 요청 메시지를 이용하여 startup시나 주기적으로 로컬 Registrar에게 현재 사용자의 바인딩 정보를 등록하며, 제3자에 의해서도 바인딩 정보를 등록할 수 있다. Registrar에서는 응답 메시지를 수신하게 되는 경우가 발생하지 않을 것이며, 만약 응답 메시지를 수신하였다면 메시지를 삭제한다.

## II. REGISTER 요청 메시지

Request-URI와 To, From, Call-ID, CSeq 헤더는 모든 REGISTER 요청 메시지에 필수적으로 있어야 하는 항목들이며, REGISTER 메시지가 어떤 용도로 사용되는가에 따라 Contact 헤더가 추가적으로 포함된다.

- Request-URI : 등록을 처리하는 도메인, 또는 Registrar 서버의 SIP URI로써 "userinfo"와 "@" component는 생략된 형태를 가진다.

- To : 등록하고자 하는 사용자의 AOR을 의미한다.

- From : 등록하는 사용자의 AOR로서 바인딩 정보가 제 3자에 의해 등록되지 않았다면 To 헤더와 동일한 값을 가진다.

- Call-ID : 한 UAC로부터의 모든 등록은 같은 Call-ID 헤더를 사용해야 한다. 같지 않을 경우 지연된 REGISTER 메시지인지 아닌지 판단할 방법이 없으므로 이 경우에는 403 (Forbidden) 응답을 전송한다.

- CSeq : CSeq 헤더값은 같은 Call-ID 가진 각각의 REGISTER 메시지마다 1씩 증가되어야 하며, Registrar

서버는 out-of-order 요청 메시지에 대해서는 에러로 처리하지 않으며 무시한다.

- Contact : Contact 헤더가 없을 경우에는, Registrar 서버에 등록된 바인딩 정보에 영향을 미치지 않으며, 다만, 현재 등록되어 있는 바인딩 정보를 알고자 할 때 사용된다. 반면, Contact 헤더에 contact address가 명시되어 있을 경우에는 등록되어 있는 정보와 일치하는지를 알아보고 등록된 정보와 다른 contact address를 바인딩 정보에 추가하고, 같으면 갱신한다.

- Expire : "expires" 파라미터는 명시된 바인딩 정보가 얼마동안 유효한지를 명시한다. "expires" 파라미터가 없을 경우에는 대신 Expires 헤더를 사용된다. 2\*\*32-1초 이하의 값을 가지며 만약, 2\*\*32-1보다 큰 값을 가지고 있을 시에는 2\*\*32-1초로 취급하며, Malformed value를 가지고 있을 시에는 3600초로 취급한다.

사용자가 다른 장소에서 다른 Call-ID값으로 등록을 할 수도 있으므로 CSeq값만으로 메시지 처리 순서를 보장할 수는 없다. 따라서 UAC는 이전에 보낸 REGISTER 메시지에 대한 최종 응답을 수신하기 전에 새로운 REGISTER 메시지를 송신해서는 안 된다. RFC 3261에서는 이전 스펙에서 사용하던 "action" 파라미터를 사용하지 않는다.

## III. SIP URI 비교

Registrar에서는 사용자의 위치정보에 대한 등록, 수정, 삭제, 조회에 대한 요청 메시지 수신시, Location Service를 제공한다. 저장되어 있는 DB에서 REGISTER 요청 메시지의 TO 헤더에 명시되어 있는 URI와 일치하는 URI를 찾아, 해당 레코드에 적힌 정보들을 이용하여 요청한 서비스를 지원한다.

이 경우, 두개의 SIP(S) URI의 동일성 여부에 대한 판단이 필요하며, 방법은 다음과 같다.

### 1. URI 스킴 비교

SIP에서는 SIP와 SIPS 두 개의 URI 스킴을 제공하며, 스킴이 일치하지 않으면 URI는 동일한 것으로 간주하지 않는다.

### 2. URI의 userinfo부분과 다른 URI component들 비교

SIP(S) URI의 userinfo 부분은 소대문자를 구분하여 비교하고, 그 외 다른 URI component들은 따로 명확하게 명시되어 있지 않으면 소대문자를 구분하지 않고 비교한다.

- 예) 동일하지 않음 (different username)
- SIP:ALICE@AtLanTa.CoM;Transport=udp
  - sip:alice@AtLanTa.CoM;Transport=UDP

3. header field나 parameter들의 순서는 일치하지 않아도 동일한 것으로 본다.

4. "reserved" set, unsafe" set 이외 문자에 대해서는 " %" HEX HEX" 으로 문자가 인코딩 되어 있다 하더라도 동일한 것으로 본다.

- 예) 동일
- sip:%61lice@atlanta.com;transport=TCP
  - sip:alice@AtLanTa.CoM;Transport=tcp

5. host name의 DNS lookup 결과로 나온 IP address는 그 host name은 동일한 것으로 보지 않는다.

- 예) 동일하지 않음
- sip:bob@phone21.bboxesbybob.com
  - sip:bob@192.0.2.4

6. 두 URI가 동일하기 위해서는 user, password, host, port component가 모두 일치되어야만 한다.

- user component (혹은 password component)를 포함하는 URI와 해당 component가 생략된 URI는 일치하지 않는 것으로 본다.
- default value를 가지는 모든 component들은 default value를 생략한 URI와 default value라 하더라도 명시한 URI는 일치하지 않는 것으로 본다.

- 예1) 동일하지 않음
- sip:bob@biloxi.com
  - sip:bob@biloxi.com:5060

7. URI uri-parameter component를 비교하는 방법

- 양쪽 URI에 모두 명시된 uri-parameter들은 반드시 일치되어야 한다.

- transport, user, ttl, method uri-parameter인 경우, 한쪽 URI에만 존재한다면 결코 동일하게 간주되지 않도록 한다.

- maddr parameter를 포함하는 URI는 maddr parameter를 포함하지 않는 URI와 동일하지 않다.

- 나머지 uri-parameter에 대해서, 만약 한쪽 URI에만 해당 uri-parameter가 존재하고 다른 한쪽에는 없을 경우, URI comparison 시 해당 uri-parameter는 무시하도록 한다.

8. URI header component는 결코 무시되지 않도록 주의한다.

- 존재하는 모든 header component는 양쪽 URI에 모두 존재해야 하고, 해당 header component는 반드시 동일해야 동일한 URI로 간주한다.

#### IV. SIP URI 비교를 위한 API 설계

ETRI에서 개발한 SIP 스택 및 Registrar 시스템에 따른 일반적인 동작을 [그림 2]를 통해 간단히 설명하도록 한다.

Registrant에서 자신의 현재 위치를 등록/조회/삭제/수정하기 위해 SIPMessage 구조체 형식으로 REGISTER 메시지를 생성하게 되면, 스트림 형태로 메시지를 인코딩 한 후에 Registrar에게 전송한다. Registrar에서는 스트림을 수신하게 되면, SIPMessage 구조체 형식으로 다시 파싱한 다음, SIPMessage 구조체 형식의 메시지를 이용하여 이후 Registrar의 동작을 수행한다.

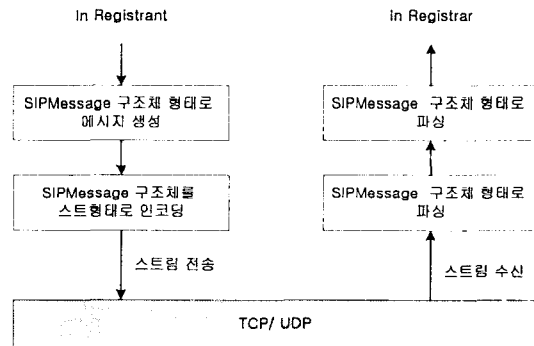


그림 2. SIP 시스템 (UA, Registrar)

```

typedef struct _URI_t {
    int Scheme;
    void *URIdata;
    char *URISTR;
} URI_t;

typedef struct _SIPURL_t {
    char * UserName;
    char * Password;
    char * HostName;
    int AddressType; // IPV4, IPV6
    int Port;
    NameValueList_t *URIParams;
    NameValueList_t *headers;
} SIPURL_t;
    
```

그림 3. URI 관련 구조체

SIPMessage구조체내 URI 값을 저장하는 필드들은 [그림 3]과 같은 URI\_t 구조체를 가진다. Scheme이라는 int 변수에는 해당 URI가 SIP인지 SIPS인지 Tel URI인지 등을 명시하게 되고, 스킴을 제외한 URI의 나머지 부분은 URIdata라는 void 포인터에 저장하게 된다. 이후, 해당 URI가 SIP이나 SIPS 스킴을 가진 경우에는 URIdata의 값을 SIPURL\_t 구조체로 매핑시키는 과정을 가지게 된다.

따라서 SIP URI를 비교하기 위한 API는 이러한 URI\_t와 SIPURL\_t 구조체 형태의 URI들을 비교하게 된다. 3장에서 살펴본 URI 비교 과정을 모두 반영하기 위해서는 다음과 같은 두 개의 API 정의가 필요하다.

1. IsSameScheme(URI\_t \* u1, URI\_t \*u2);

u1->Scheme와 u2->Scheme가 일치하는지 체크한다. 만약 일치하면, IsSameSIPURL 함수를 호출한다.

2. IsSameSIPURL(SIPURL\_t \* u1, SIPURL\_t \*u2);

3장에서 설명한 URI 비교 규칙을 하나씩 점검한다. 일치 조건을 모두 만족한 경우에만 두 개의 URI가 동일한 것으로 본다.

위에서 정의한 두 개의 API는 앞에서 설명한 것과 같이 Registrar에서만 사용되는 것은 아니며, UA, Proxy Server, Redirect Server 등에서도 여러 용도로 다양하게 사용되어진다.

## V. 결 론

인터넷 사용자 수의 급격한 증가로 인터넷 서비스 보급이 대중화됨에 따라 인터넷 텔레포니 기술이 전기통신사업자 및 인터넷 서비스 사업자들의 주요 이슈가 되고 있다. 현재 대부분의 인터넷 전화 서비스는 ITU-T에서 제정한 H.323 표준에 따라 제공되고 있으나 최근에는 IETF에서 개발한 SIP 프로토콜과 같은 차세대 VoIP 표준 기술을 기반으로 한 제품 개발을 추진하고 있는 추세이다.

SIP 프로토콜은 인터넷 텔레포니 서비스를 위하여 IETF에서 여러 번의 개정 작업을 거침에 따라 콜 시그널링 프로토콜로 적용하는데 무리가 없을 것으로 예상된다. 따라서 인터넷상에서의 음성 서비스 제공을 위한 여러 형태의 제품 개발 시 SIP 프로토콜은 기본이 될 것이다. SIP 프로토콜에서 개인의 이동성을 보장하기 위해서는 사용자들의 위치가 변경될 때마다 현재 위치를 Registrar에게 등록 시켜야 하며, 이러한 등록 정보들은 이후 Proxy 서버나 Redirect 서버를 위한

Location Service 제공을 위해 사용된다. 본 논문을 통해 2002년 7월 SIP 표준으로 제정된 RFC3261에 따른 Registrar의 기본동작을 살펴보았으며, 두 SIP URI의 동일성 여부를 판단하기 위해 방법에 대해 분석하고 관련 API들에 대해 살펴보았다.

## 참고문헌

- [1] J. Rosenberg, H. Schulzrinne, et al., "SIP: Session Initiation Protocol," RFC3261.
- [2] 박선옥, 강신각, "SIP Registrar 기능 분석 및 설계," 2002 정보과학회 추계학술발표회.
- [3] Wook Hyun, et.al, "Design & Implementation of SIP Proxy Server," ICACT 2002.
- [4] www.ietf.org, IETF homepage.