NExpert.net

Session Initiation Protocol 의 이해

Session Initiation Protocol 의 이해와 활용

Written by 라인하트.
Edited by 허클베리 핀
1/23/2009

Chapter 1

RFC 3261 분석

1. RFC 3261 분석

SIP 는 차세대 VoIP 프로토콜이며, 현재의 VoIP 의 대세는 H.323 에서 SIP 로 바뀌는 중입니다. 대부분의 VOIP 장비는 H.323 표준을 지원하고 있으며, 호환성이 뛰어나서 대부분의 H.323 장비들 간의 연동은 잘 이루어지고 있습니다. 그러나, 현재 VoIP 시장의 화두는 Unified Communications(이하 UC)이며, 이는 단순히 전화 또는 영상의 전달만이 목적이 아니라, 기존의 모든 통신 수단을 하나로 통합하고자 하는 진화의 방향입니다.

UC 를 구현 하는데 있어 H.323 은 한계가 있습니다. 따라서, 처음부터 IP 관련 표준을 선도해 온 IETF (http://www.ietf.org) 에서 표준화된 SIP 는 UC 의 Protocol 로써 최상의 조건을 가지고 있다고 볼 수 있습니다. 이러한 SIP 에 대해 자세히 살펴보기 위해서는 RFC 3261 를 참조해야 합니다.

RFC 3261 는 하나 또는 그 이상의 참가자와의 통신과 세션의 생성, 변경, 종료에 대한 application layer 프로토콜인 SIP 에 대해 설명한 권고안입니다.

RFC 3261 의 Overview of Operation 은 SIP 를 쉽게 이해할 수 있도록 되어 있어 이 부분을 정리를 하였습니다

1.1. Overview of SIP Functionality

SIP 는 멀티미디어 통신을 생성하고 종료하기 위한 5 가지 요소는 다음과 같습니다.

- **User Location:** 통신에 참가할 단말을 결정
- **User Availability:** 통신에 참여할 수신측의 통화 가능여부 결정
- **User Capabilities:** 통신간에 사용될 미디어 및 미디어 파라미터 결정
- **Session Setup:** 수신측 및 송신측에 세션 파라미터 생성
- **Session Management:** 세션의 종료, 전환, 세션 파라미터 변경, 부가 서비스 연동

SIP 는 5 가지 요소의 기능을 통해 멀티미디어 통신을 가능하게 하며, SIP 은 UA(User Agent), Proxy Server, Redirect Server, Registrar 등의 개체들로 이루어져 있습니다.

- **UA (User Agent)**
  접속요청 메시지를 송신할 때에는 클라이언트 형식(UAC, User Agent Client)으로 동작하고, 접속요청 메시지를 수신하여 처리할 때에는 서버 형식(UAS, User Agent Server)으로 동작합니다. UA 는 다른 UA 와 직접 연결을 설정하거나 Proxy/Redirect Server 들의 도움으로 다른 UA 와 연결을 설정하며, 호(呼, call) 상태를 저장하고 관리합니다.

- **Proxy Server**
  UA 로부터의 수신한 접속 요청 메시지를, 다른 도메인(domain)의 Proxy 혹은 Redirect Server 로 전달하거나, 해당 도메인 내의 UA 로 전달하는 기능을 수행하고 과금(billing)을 위한 정보들을 유지합니다.
- **Redirect Server**
  수신한 접속 요청 메시지를 다른 UA 나 Proxy Server 에게 직접 전달하지 않고, 접속 요청 메시지를 보내 온 해당 UA 나 Proxy Server 에게 그들이 접속 요청 메시지를 재전송해야 할 UA 나 Proxy Server 의 주소를 알려 주는 역할을 합니다.
- **Registrar**
  UA 로부터 등록 요청 메시지를 수신하고 이를 SIP 이 아닌 다른 별도의 프로토콜을 이용하여 Location Service 를 제공하는 시스템에 저장합니다. (Location Service 는 이 정보를 Proxy 혹은 Redirect Server 에 제공하여 접속 요청이 잘 전달될 수 있도록 합니다.)

SIP 는 완벽한 멀티미디어 아키텍쳐를 구성하기 위해 다른 IETF 프로토콜과 함께 사용되는 컴포넌트라고 볼 수 있습니다. 멀티미디어 아키택쳐는 다음과 같은 프로토콜을 포함하고 있습니다.

- RFC 1889 Real-Time Protocol (RTP)
  실시간 데이터 전송 및 QoS 에 대한 피드백 제공
- RFC 2326 Real-Time Streaming Protocol (RTSP)
  스트리밍 미디어 전송을 제어
- RFC 3015 Media Gateway Control Protocl (MEGACO)
  Public Switched Telephone Network(PSTN)과 IP 네트워크간의 연동을 위한 게이트웨이 제어
- RFC 2327 Session Description Protocol (SDP)
  멀티미디어 세션 파라미터 정의

위와 같이, SIP 는 IETF 의 멀티미디어 아키택쳐 가운데 하나로 사용자에게 완벽한 서비스를 제공하기 위해서는 다른 프로토콜과 결합하여 사용되어야 합니다.

SIP 는 서비스를 제공하지 않고, 서비스를 구현하기 위해 사용될 Primitives (매개 변수)를 제공합니다. 예를 들면, "발신자 정보 표시 서비스"가 구현될 때, SIP 가 서비스를 제공하는 것이 아니라 Primitives 에 의해 단순히 SDP 에 의해 세션 정보를 전송할 뿐입니다. 따라서, 이 서비스가 이 값을 이용하여 구현하는 것입니다. 따라서, 이 Primitives 는 여러 다른 서비스에 의해 사용될 것입니다.

1.2. Overview of Operation - Basic Call Flow

다음의 간단한 예를 통해 살펴보겠습니다. 앨리스는 소프트폰을 가지고 있으며, 밥은 SIP 전화기를 가지고 있습니다. 앨리스가 밥에게 전화를 거는 상황이며, 최소한의 파라미터만을 가지고 교환하는 것으로 합니다.

Figure 1: SIP session setup example with SIP trapezoid

앨리스는 atlanta.com 의 Proxy Server 로 INVITE 메시지를 아래와 같이 전송합니다 .첫 줄은 INVITE 메소드와 URI (Uniform Resource Identifier)를 나타내고 있으며, 메일 주소 체계와 비슷합니다.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

(Alice's SDP not shown)
```

- **Via**
  앨리스가 INVITE 요청에 대한 응답을 받고자 하는 주소를 나타냅니다.
- **To**
  Display Name 인 "Bob"과 실제 연결되어야 할 SIP URI 를 나타냅니다.
- **From**
  Display Name 인 "Alice"와 실제 요청한 SIP URI 를 나타냅니다.
- **Call-ID**
  이 호를 위한 global unique identifier 를 사용하며, 일반적으로 호스트 네임 또는 IP address 와 랜덤 스트링을 결합하여 생성됩니다. 따라서, To/ From/ Call-ID 가 결합으로 앨리스와 밥간의 Pee-to-peer SIP 관계를 정의합니다.
- **Cseq**
  Commnad Sequence 는 정수와 메소드 이름을 포함합니다. 새로운 요청마다 증가합니다.
- **Contact**
  SIP URI 를 포함하며, 앨리스에게 접근할 수 있는 직접적인 경로를 나타냅니다. 일반적으로 FQDN (Fully qualified domain name)을 사용합니다. 그러나 도메인 네임을 DNS 에 등록하지 않은 경우에는 IP address 를 사용하기도 합니다. Via 헤더 필드가 요청에 대한 응답 경로를 나타내고, Contact 헤더 필드는 미래의 요청을 보낼 경로를 말합니다. 조금 어렵게 되어 있는 데요. 요청에 대한 응답은 Via 헤더 필드를 참조하며, 신규 요청을 생성할 경우는 Contact 헤더 필드를 참조한다는 것입니다.

미디어 타입, 코덱, 샘플링 속도 등은 SDP(Session Description Protocol)를 통해 전송됩니다. (참조: RFC 2327 SDP 혹은 NExpert.net 의 "폴리콤 PVX 와 HDX 간의 SIP 화상 통신 패킷 분석")

INVITE 메소드는 atlanta.com 의 SIP Proxy Server 로 전송됩니다. 실제 앨리스는 밥의 위치를 알 수가 없습니다. Proxy Server 는 요청을 받은 후에 100 Trying 메시지를 앨리스에게 전송하며, 이 메시지의 의미는 정확하게 앨리스의 INVITE 메소드를 수신했으며, 이 메시지를 밥에게 보내기 위해 처리중임을 나타냅니다. atlanta.com 의 SIP Proxy Server 가 biloxi.com SIP Proxy Server 로 메시지를 전송하기 전에 INVITE 메시지내의 Via 헤더 필드에 자신의 주소를 추가합니다.

biloxi.com 의 SIP Proxy Server 는 INVITE 메시지를 수신 후 100 Trying 을 전송하고, 밥의 IP address 를 확인한 후 Via 헤더 필드에 자신의 주소를 추가에게 밥에게 전송합니다.

INVITE 메시지를 수신 후 밥의 전화기는 링을 울리게 되고, 180 Ringing 을 biloxy.com 의 SIP Proxy Server 로 전송하여 역방향으로 메시지가 앨리스에게 전송되며, 앨리스는 Ringback tone 을 듣게 됩니다.

밥이 수화기를 들게 되면, 200 OK with SDP 메시지가 앨리스에게 전송됩니다.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com
   ;branch=z9hG4bKnashds8;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
   ;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com
   ;branch=z9hG4bK776asdhds ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131

(Bob's SDP not shown)
```

200 OK 메시지를 살펴보면, Via, To, From, Call-ID, CSeq 헤더 필드는 INVITE 메시지로부터 복사합니다. Contact 필드는 밥에 의해 변경되어 미래의 요청에 사용될 것입니다.

앨리스의 전화기는 200 OK 를 수신 후 Ringback tone 생성을 중지하고, 밥이 수화기를 들었음을 알립니다. 앨리스의 전화기는 ACK (acknowledgement) 메시지를 송신합니다. 위의 Call Flow 를 다시 보시면, 앨리스 전화기의 200 OK 메시지가 직접 밥의 전화기로 전송되는 것을 알수 있습니다. 앨리스와 밥의 전화기는 Contact 헤더 필드를 통해 서로 상대방의 주소를 교환하였기에 발생합니다. 두 개의 SIP Proxy Server 는 더 이상 호를 추적할 필요가 없으므로, Call Flow 에서 제외되는 것입니다. 이것이 SIP 세션 설정을 위한 "three-way handshake"입니다.

이제 앨리스와 밥은 SDP 에 의해 협상된 파라미터를 통해 미디어 세션이 시작될 것입니다. 일반적으로 이 또한 앨리스와 밥간의 직접 교환되며, Proxy Server 는 제외됩니다.

만일, 미디어 세션에 대한 파라미터 변경이 필요할 경우 re-INVITE 메시지가 전송되고, 상대방은 200 OK 를 전송하여 새로운 협상에 동의함을 표시하고, 동의하지 못할 경우 488 (not acceptable here) 에러를 전송합니다. 그러나, re-INVITE 의 fail 은 기존에 연결된 호에 대해서는 영향을 미치지 않습니다.

만일, biloxi.com Proxy Server 가 단말간의 SIP Signaling 메시지를 모두 확인하고 싶어할 경우, 즉, 메시지가 SIP Proxy Server 를 경유하도록 하고 싶다면, INVITE 메시지내의 Record-Route 라우팅 헤더 필드를 추가할 수 있습니다. mid-call feature 를 제공하는 SIP Proxy Server 에 유용하게 사용됩니다.

1.3. Overview of Operation - Registration

SIP 에 있어서 등록과정은 필수적인 요소이다. 기본적으로 Peer-to-peer Protocol 이므로 SIP Proxy Server 는 옵션이지만, 단말의 숫자가 증가할 경우 모든 경로에 대한 정보를 단말이 보유하는 것은 불가능하므로 일반적으로 SIP Proxy Server (일반적으로, Registra Server 와 함께 구현됨)를 사용하며, 단말들은 등록과정을 진행합니다. 등록과정을 통해 SIP Proxy Server 는 단말의 현재 위치를 확인할 수 있습니다. 즉, address-of-record URI 와 Contact address 를 바인딩하는 것입니다.

단말은 REGISTRATION 메시지를 SIP Proxy Server 에 전송하여 200 OK 를 수신하는 것으로 등록 과정이 진행됩니다. 등록 시 밥은 집전화와 사무실 전화를 동시에 등록할 수 있으며, 한 명 이상의 사용자가 동시에 하나의 전화기에 등록될 수 있습니다.

Registration 메시지에는 다음과 같은 헤더 필드를 포함합니다

- **Request-URI**
  등록 시에 사용되는 위치서비스의 도메인으로 "sip:chicago.com" 으로 표시된다. SIP URI 의 @이나 userinfo 가 표시되어서는 안됩니다.
- **To:**
  address-of-record 는 SIP URI 로 표시되어야 하며, 등록 신청,변경, 의뢰에 대한 address-of-record 를 포함합니다.
- **From:**
  등록에 대한 응답되는 address-of-record 를 포함하며, 일반적으로 To 헤더 필드와 내용이 동일하다.
- **Call-ID**
  하나의 UAC 로 부터의 모든 등록 메시지는 같은 Call-ID 를 가집니다.

Capability Negotiation

Capability Negotiation 는 SDP 를 통해 이루어집니다. INVITE with SDP 로 밥이 앨리스에게 전송하면, 앨리스는 밥에게 200 OK with SDP 로 응답하여 two-phase 협상이 이루어 집니다. 이 과정을 통해 기본적인 협상이 성립됩니다. 즉, 앨리스가 제시하고, 밥이 응답하는 형식을 취하는 것입니다.

Chapter2

Session Initiation Protocol - RFC 3261

2. Session Initiation Protocol – RFC 3261

최근 UC 분야의 프로토콜 중 화두는 SIP 입니다. H.323 과 MGCP 는 Call Server 와 Gateway 간에 주로 사용이 되고, Line side 는 SIP 가 대세입니다. 지인들의 경우에도 SIP 프로토콜을 이용하여 IP Phone 또는 Gateway 를 사용하지만, Architecture 에 대한 이해가 부족하여 장애나 이 기종 장비간 연동 시에 어려움을 겪는 것을 보았습니다. 사실 저도 편하게 글을 써보고자 검색엔진에서 SIP 관련 자료를 찾아보았지만, 체계적으로 정리된 자료가 없는 것이 아쉬웠습니다. 이 연재를 통해 UC 현업에 계시거나, UC 를 공부하시는 분들에게 도움이 되길 기대합니다.

이 글은 "Cisco Networkers"에서 발표되었던 SIP 관련 Presentation 파일 몇 개와 IETF 의 RFC 문서를 기초로 작성하였습니다.

2.1. SIP 의 개요

SIP 는 Session Initiation Protocol 의 약자로 응용계층의 호 Signaling Protocol 입니다. SIP 는 하나 또는 그 이상의 참가자와 멀티미디어 세션의 생성, 변경, 종료에 대해 정의하였습니다. SIP 에서 세션은 다음과 같이 정의할 수 있습니다.

- Internet multimedia conferences (다자간 회의)
- Internet telephone calls (음성 전화)
- Internet video sessions (영상 전화)
- Multimedia distribution (멀티미디어 분배)
- Subscriptions and Nofitifations for Events (이벤트 신청 및 통지)
- Publications of State (상태 정보 배포)

SIP 상에서 세션이라는 것이 전화를 걸고 받기 위한 시그널링이라고 단순히 생각할 수 없습니다. 현재, SIP 를 통해 단순히 호 정보만을 교환하는 것이 아닌 다양한 곳에서 폭넓게 사용되고 있습니다.

2.2. SIP 의 History

SIP 는 1999 년 3 월 RFC 2543 SIP 가 처음 IETF 에서 표준화되었으며, 2002 년 7 월 RFC 3261 SIP 로 개정되었습니다. 또한, SIP 상의 기본적인 내용만 정의되었던 RFC 3261 를 보완한 45 개의 추가적인 SIP 관련 표준이 표준화 되어 다양한 부가서비스를 개발할 수 있게 되었습니다. 물론, 이외에도 다수의 SIP 관련 Draft 문서들이 있습니다. RFC 3261 는 SIP 를 알기 위한 기본적인 문서이지만, 실제 업무에서 부딪히는 많은 부분은 아래 명시된 자료들과 함께 확인해야 합니다.

국내 및 해외 SIP 장비 중에서도 아래 모든 표준을 지원하고 있는 장비는 없는 듯합니다. 그러나, 차츰 시간이 지나면서 모두 지원될 수 있을 것으로 기대합니다.

- RFC 2976 The SIP INFO Method
- RFC 3204 MIME media types for ISUP and QSIG Objects
- RFC 3261 SIP
- RFC 3262 Reliablity of Provisional Responses in SIP
- RFC 3263 SIP : Location SIP Servers
- RFC 3265 SIP-Specific Event Notification
- RFC 3311 SIP UPDATE Method
- RFC 3312 Integration of Resource Management
- RFC 3313 Private SIP Extensions for Media Authorization
- RFC 3319 DHCPv6 Options for SIP Servers
- RFC 3323 A Privacy Mechanism for the SIP
- RFC 3325 Private Extensions to the SIP for Asserted Identity within Trusted Networks
- RFC 3326 The Reason Header Field for the SIP
- RFC 3327 SIP Extension for Tedistering Non-Adjacent Contacts
- RFC 3329 Security Mechanism Agreement for the SIP
- RFC 3310 HTTP Digest Authentication Using Authentication and Key Agreement
- RFC 3361 DHCP Option for SIP Servers
- RFC 3420 Internet MEdia Type message / sipfrag
- RFC 3428 SIP for Instant Messaging
- RFC 3515 The SIP Refer Method
- RFC 3581 The SIP Extension for Symmetric Response Routing
- RFC 3608 SIP Extension Header Field for Service Route Discovery During Registration
- RFC 3853 S/MIME AES Requirement for SIP
- RFC 3840 Indicating User Agent Capabilities in the SIP
- RFC 3841 Caller Preferences for SIP
- RFC 3891 The SIP 'Replaces' Header
- RFC 3892 The SIP Referred-By Mechanism
- RFC 3893 SIP Authenticated Identity Body (AIB) Format
- RFC 3903 An Event State Publication Extension to the SIP
- RFC 3911 The SIP 'Join' Header
- RFC 3968 The IANA Header Field Parameter Registry for the SIP
- RFC 3969 The IANA Universal Resource Identifier (URI) Parameter Registry for the SIP
- RFC 4032 Update to the SIP Preconditions Framework
- RFC 4028 Session Timers in the SIP

- RFC 4092 Usage of the SDP Alternative Network Address Types (ANAT) Semantics in the SIP
- RFC 4168 The SCTP as a Transport for SIP
- RFC 4424 The History-Info Header for SIP
- RFC 4411 The Reason Header for Preemption for SIP
- RFC 4412 The Communications Resource-Priority Header for SIP
- RFC 4488 Suppression of SIP REFER Method Implicit Subscription
- RFC 4508 Conveying Feature Tags with SIP REFER Method
- RFC 4485 Guidelines for Authors of Extensions to the SIP
- RFC 4483 A Mechanism for Content Indirection in SIP Messages
- RFC 4538 Request Authorization through Dialog Identification in the SIP
- RFC 4474 Enhancements for Authenticated Identity Management in the SIP

2.3. SIP 가 이용하는 IETF 프로토콜

SIP 는 Signaling 에 대한 정의이며, 이외의 다른 프로토콜과 연계하여 사용됩니다.

- RFC 2616 Message formatting (HTTP 1.1)
- RFC 4566 Media Description (SDP)
- RFC 3550 RTP
- RFC 1738 URL & RFC 2396 URI
- RFC 1034 & RFC 1035 DNS
- RFC 2131 Device Mobility
- RFC 2045 MIME (Multiperpose INternet Mail Extensions)
  메시지 포맷 (US-ASCII)에 대한 정의로 SIP 의 Message Body 에 사용됩니다.
- RFC 4301 & RFC 4303 IPSec
  보안관련 규정입니다.
- RFC 4346 TLS
  Signaling 에 대한 보안 규정입니다.

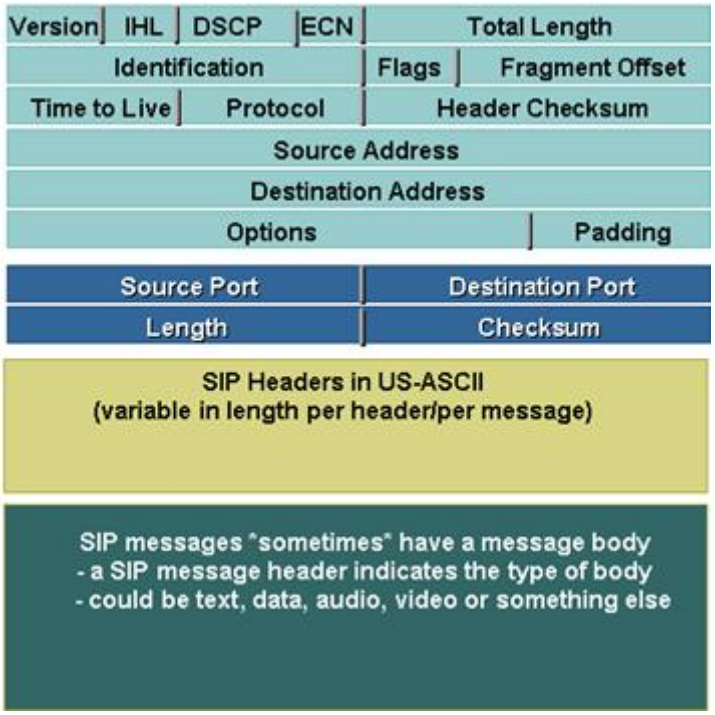
2.4. SIP Components 및 SIP Header 설명

Chapter 1 의 "RFC 3261 SIP 분석" 이라는 글에 설명을 참조하시기 바랍니다.

2.5. SIP 패킷의 구조

아래 그림은 일반적인 SIP 를 포함한 패킷의 구조이며, IP Header (20Byte)/UDP Header (8Byte)/SIP Header/ SIP Message Body 로 이루어져 있습니다. SIP 헤더는 Chapter 1 의 "RFC

3261 SIP 분석" 이라는 글에 자세히 언급되어 있으며, SIP Message Body 는 있을 수도 있고, 없을 수도 있는 옵션이며, 주로 SDP 의 내용이 첨가된다고 보면 되겠습니다.



위의 그림에서는 Transport Layer 프로토콜로 UDP 의 사용이 일반적이지만, TCP 또는 SCTP 가 사용될 수 있습니다. SCTP 는 잘 사용되지 않습니다만, Sigtran 프로토콜이 사용하는 것이 일반적입니다. SCTP 는 TCP 와 UDP 의 장점을 결합해 놓은 것으로 IPv6 에서만 기본 스택으로 정의되어 있기 때문에 IPv6 에서 많이 사용될 것으로 생각됩니다. 나중에 Sigtran 을 연재할 기회가 생기면 그 때 다시 언급하도록 하겠습니다. SCTP 만 가지고도 글 한 두개는 만들 수 있을 만큼 방대한 내용입니다.

Chapter 3

Session Description Protocol

3. Session Description Protocol

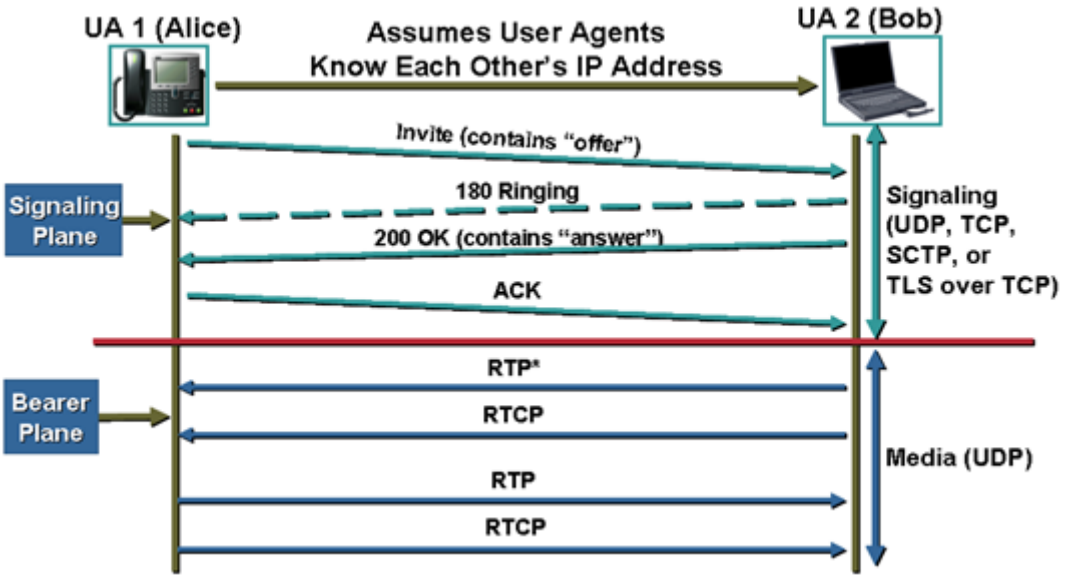
3.1. SDP 의 개요

SDP 는 Session Description Protocol 로 멀티미디어 세션 파라미터를 설정합니다. 즉, H.323 프로토콜의 H.245 와 같은 기능을 수행합니다. SDP 는 현재 RFC 2327 을 개정한 RFC 4566 이 표준화 되었습니다. 또한, SIP 뿐만 아니라 MGCP / Megaco 에서도 멀티미디어 세션 파라미터 설정을 위해 SDP 를 사용합니다.

SDP 를 이해해야지만, 이기종 장비간 연동이나 장애처리의 많은 부분을 해결할 수 있습니다. 이 글에서는 SDP 4566 의 기본적인 부분과 RFC 3264 를 기준으로 이야기를 전개하겠습니다. 실제 업무환경에서 필요한 내용 위주의 내용으로 쓰도록 하겠습니다.

3.2. Offer/Answer Model

RFC 3264 An Offer/Answer Model with the SDP 를 통해 Capability Exchange 를 설명합니다. SIP 도 마찬가지로 Request / Response Model 이므로 같은 방식으로 동작합니다. 이 Offer / Answer Model 이라고 해서 특별한 것은 아닙니다. 아래 그림을 참조하시기 바랍니다.



위 그림은 SIP Proxy Server 가 없는 것으로 가정한 것입니다. Invite 메시지에 SDP 메시지가 포함되어 전송되고, 이때 Alice 의 사용 가능한 Capability 가 Offer 됩니다. 200 OK 메시지에 SDP 에는 Bob 의 사용 가능한 Capability 가 Answer 되고, RTP 가 개방될 수 있습니다.

3.3. SDP Lines Message 설명

SDP 도 SIP 마찬가지로 텍스트 기반이며, 다음의 내용을 포함합니다.

- v = protocol version
- o = owner/creator and session identifier)
- s = session name
- c = connection information – not required if included in all media
- k = encryption keys
- t = time the session is active
- m = media description and transport address
- a = (zero or more) media attributes lines

위의 내용에서 관심을 가질 사항은 딱 두 개입니다 "m"과 "a"입니다. 실제 사용될 코덱과 코덱의 속성에 대해 설명합니다.

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.com
c=IN IP4 10.1.3.33
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

위의 그림은 SDP 의 일반적인 메시지 포맷을 나타낸 것입니다. "m"의 내용을 보면 Media 는 음성이며 UDP port 49172 를 사용하고, 코덱은 0 를 쓴다고 되어 있습니다. 코덱 0 은 a=rtpmap 에 표시되어 있습니다. PCMU/8000 이라고 되어 있네요 1 초를 1/8000 로 샘플링 한 G.711ulaw 코덱을 사용하자고 보냅니다. 이런 것이 SDP 의 역할입니다. 좀 깊게 알고 싶으신 분은 "SIP 화상회의 패킷 분석" 자료를 참조하시기 바랍니다.

3.4. RFC 3264 의 기본 예제

RFC 3264 에는 SDP 교환에 대한 예제가 나와 있습니다. 이를 읽어보시면, Media capability 에 대한 교환과 표시를 쉽게 이해할 수 있습니다. 개인적으로는 상당히 잘 정리되고, 쉽게 이해할 수 있는 RFC 문서라고 생각됩니다. 이 부분의 내용은 RFC 3264 의 10.1 절에 나와 있는 내용입니다.

Caller 인 앨리스가 Callee 인 밥에게 Offer 를 시도합니다.

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.anywhere.com
s=
c=IN IP4 host.anywhere.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVP 31
a=rtpmap:31 H261/90000
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

위의 내용을 보면, 하나의 양방향 Audio Stream 과 두 개의 양방향 Video Stram 을 앨리스가 요구합니다. 세부적인 내용은 설명하지 않아도 되리라 생각합니다. 이에 아래처럼 밥이 Answer 를 합니다.

```
v=0
o=bob 2890844730 2890844730 IN IP4 host.example.com
s=
c=IN IP4 host.example.com
t=0 0
m=audio 49920 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

앨리스의 Offer 와 어떤 차이가 있는 지 보시면, 첫 번째 Video 에 대한 속성 "a=rtpmap:31 H261/900000" 라는 부분이 없습니다. 즉, 밥은 첫 번째 영상을 받고 싶어하지 않습니다. 따라서, 음성과 두 번째 영상만을 서로 주고 받을 것입니다.

이 시점에서 밥은 앨리스에게 음성 UDP 포트를 49920 에서 65422 로 변경을 Offer 합니다. 헐 예제를 만들려니까 아주 복잡하게 만듭니다. 또한, 수신 전용의 이벤트 처리를 위한 RTP Payload Type 110 인 음성 채널 하나를 더 요청합니다. 이 음성 채널은 실제 음성을 교환하는 것이 아니라 이벤트 처리만을 담당하게 될 것입니다.

```
v=0
o=bob 2890844730 2890844731 IN IP4 host.example.com
s=
c=IN IP4 host.example.com
t=0 0
m=audio 65422 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
m=audio 51434 RTP/AVP 110
a=rtpmap:110 telephone-events/8000
a=recvonly
```

이 Offer 에 앨리스는 다음과 같이 Answer 합니다. 밥의 요청을 모두 수락한 것입니다. 밥의 수신 전용이므로, 앨리스의 송신 전용 음성 채널 개방이 합의 되었습니다.

```
v=0
o=alice 2890844526 2890844527 IN IP4 host.anywhere.com
s=
c=IN IP4 host.anywhere.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 31
a=rtpmap:31 H261/90000
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
m=audio 53122 RTP/AVP 110
a=rtpmap:110 telephone-events/8000
a=sendonly
```

3.5. RFC 3264 의 제공된 다수의 코덱에서 선택하기

일반적으로 IP Phone 이나 Gateway 의 경우 DSP 를 사용합니다. SIP 단말의 지원 코덱은 DSP 에 전적으로 의존하게 됩니다. 아래 그림처럼 앨리스는 밥에게 하나의 음성 채널을 개방하며, 사용 가능한 코덱은 G.711ulaw, G.723, G.729 3 개임을 나타내며,선호하는 코덱은 G.711ulaw 임을 나타내기 위해 순차적으로 나열해 보냅니다. "a=inactive"는 코덱이 선택되기 전까지는 Media 를 받아들일 수 없음을 의미합니다.

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.anywhere.com
s=
c=IN IP4 host.anywhere.com
t=0 0
m=audio 62986 RTP/AVP 0 4 18
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000
a=rtpmap:18 G729/8000
a=inactive
```

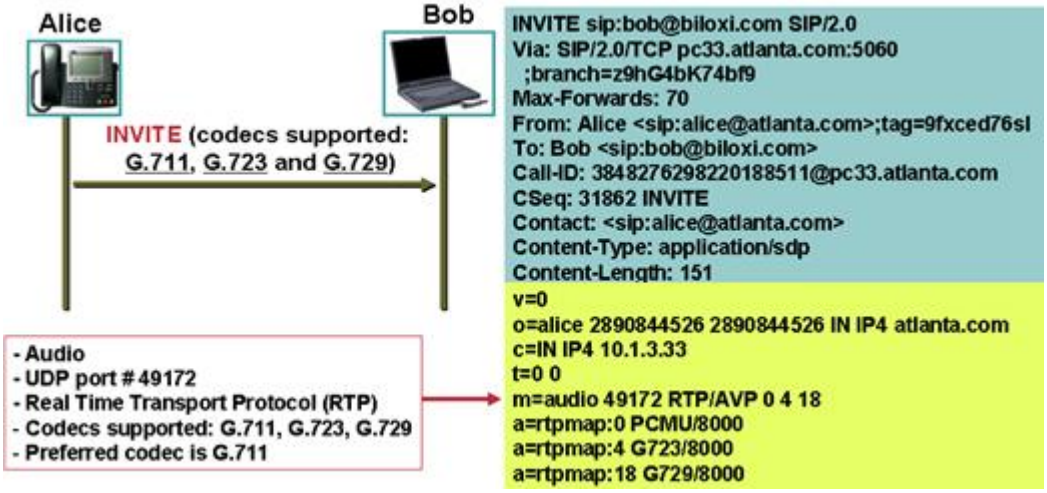
밥은 앨리스에게 사용 가능한 코덱은 2 개라고 보내며, 선호하는 것은 G.711ulaw 라고 보냅니다. 여전히 inactive 상황입니다.

```
v=0
o=bob 2890844730 2890844731 IN IP4 host.example.com
s=
c=IN IP4 host.example.com
t=0 0
m=audio 54344 RTP/AVP 0 4
a=rtpmap:0 PCMU/8000
a=rtpmap:4 G723/8000
a=inactive
```

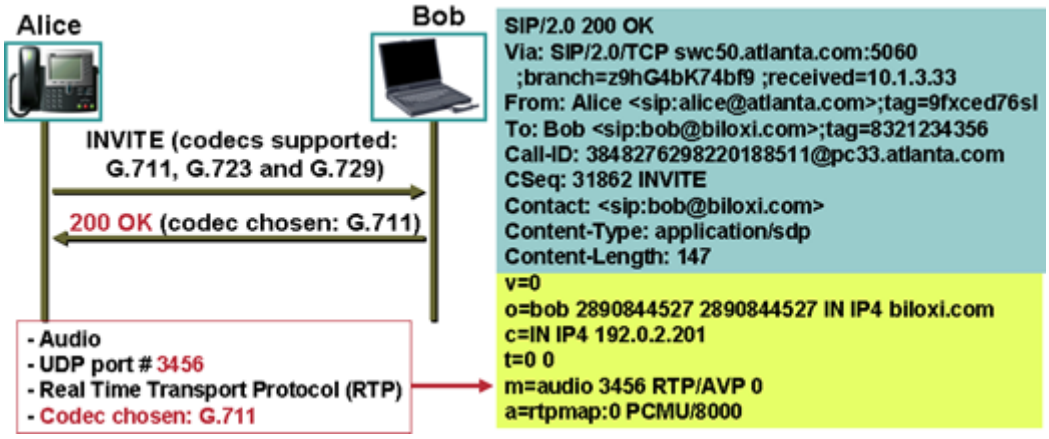
이 때에는 서로 새롭게 코덱을 선택하게 되고, inactive 상태를 sendrecv 로 전환하도록 다시 한번 Offer 와 Answer 를 교환합니다. 일반적으로는 Offer 또는 Answer 상의 맨 상위 코덱을 사용하는 것이 일반적입니다.

3.6. 일반적인 SIP 시그널링 예제

지금까지 SDP 의 입장에서만 살펴보았습니다. 전체적인 프로세스 상에서 어떻게 움직이는 지 확인하겠습니다.



앨리스는 Invite with SDP 로 Offer 합니다. 내용은 단순합니다. 음성, UDP Port 49172, 지원 코덱(G.711ulaw, G.723, G.729) 그리고 이중에서 G.711ualw 를 선호함을 나타냅니다.



밥은 200 OK with SDP 로 음성, UDP Port 3456, 선호 코덱 G.711ulaw 로 보냅니다. 코덱은 선택되었으며, 반드시 앨리스에 의해 ACK 되어야 합니다.


```
ACK sip:bob@192.0.2.201 SIP/2.0
Via: SIP/2.0/TCP pc33.atlanta.com:5060
 ;branch=z9hG4bK74bd5
Max-Forwards: 70
From: Alice <sip:alice@atlanta.com>;tag=9fxced76sl
To: Bob <sip:bob@biloxi.com>;tag=8321234356
Call-ID: 3848276298220188511@pc33.atlanta.com
CSeq: 6187 ACK
Content-Length: 0
```

이 장에서는 Invite with SDP 를 살펴보았습니다. 그러나, 경우에 따라서는 Invite without SDP 가 많은 경우에 발생합니다. 이러한 경우에는 SDP 교환은 어떻게 발생할까요? ^^ 다음 챕터에서 살펴봅시다.

Chapter 4

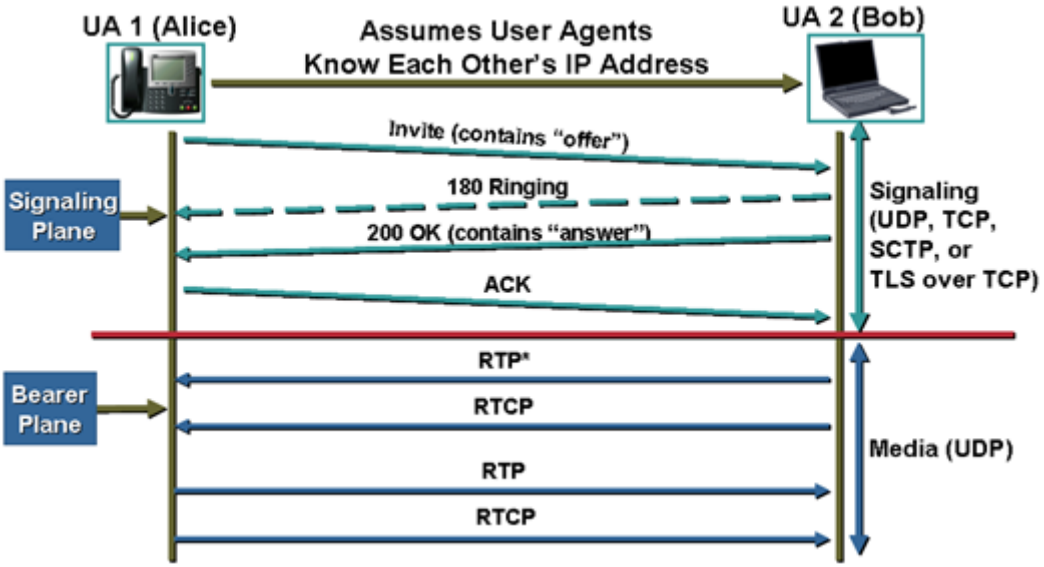
Early Media in Session Description Protocol

4.Early Media in Session Description Protocol

SDP 를 마무리 짓기 위해 이 부분을 짚고 넘어갑니다. Early Media 를 굳이 언급하지 않아도 그 의미를 쉽게 이해하시겠지만, RFC 3959 The Early Session Disposition Type for the SIP 와 RFC 3960 "Early Media and Ringing Tone Generation in the SIP" 의 내용을 위주로 정리해 보겠습니다. 표준안을 아는 것과 모르는 것의 차이는 크다는 것을 믿기 때문입니다.

4.1. Early Media 의 개요

아래 그림은 일반적인 SIP 호 설정 시나리오입니다. 앨리스는 수화기를 들어 전화번호를 누르면 Invite 메시지와 함께 Offer 가 이루어집니다. 밥의 전화기는 Ringing (따르릉)하게 되며, 180 Ringing 메시지가 앨리스에게 전달됩니다. 180 Ringing 을 받은 앨리스는 Local Ringback 을 들으면서, 밥의 전화기가 울리고 있다고 인지합니다. 밥은 Ringing 소리를 듣고 전화가 왔음을 인지하게 되어 전화기로 다가가 수화기를 드는 순간 200 OK 메시지와 Answer 가 앨리스에게 전달됩니다. 200 OK 를 받은 앨리스는 ACK 를 밥에게 전송하며, 통화가 시작됩니다.



일반적인 SIP 통화 시나리오 상에서 두 가지 문제가 발생하게 됩니다.

- **Remote Ring back**
  180 Ringing 메시지를 전달받은 앨리스의 전화기는 자체적으로 Ring back tone 을 발생시킵니다. 그러나, 현실에서는 컬러링이나 Remote 링백톤을 들을 수 있어야 합니다.
- **Media Clipping**
  보통 사람들은 수화기를 들면서 "여보세요"라고 말합니다. 위의 시나리오는 200 OK 를 보내고 ACK 를 받을 때까지의 이야기는 전달될 수 없게 됩니다. 일반적으로 SIP Signaling 은 몇 개의 Proxy 를 거쳐서 전달되지만, RTP 는 전화기간에 바로 전달이

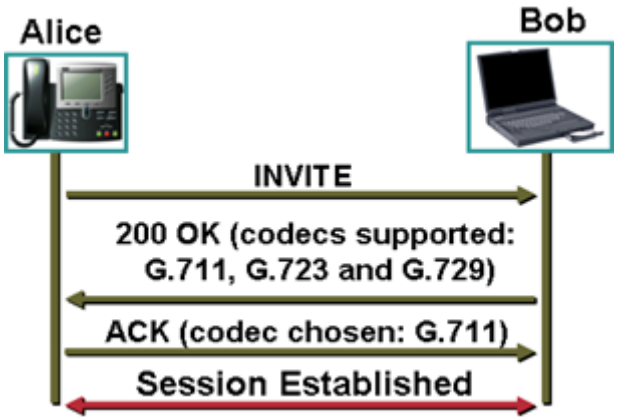
됩니다. 따라서 Media Clipping 이 발생할 수 밖에 없어 밥의 "여보세요"가 앨리스에게는 "세요"라고 들릴 것입니다.

이러한 문제점은 SDP 의 일반적인 Offer / Answer 절차 이전에 Media 가 전송되어야 한다는 것을 의미합니다. 즉, Early Media 는 정규 Offer / Answer 절차 이전에 전송되는 RTP 를 가리킵니다. 그렇다면, Early Media 의 발생시점은 Invite 의 생성에서부터 ACK 신호까지입니다. ACK 이후에는 정상적인 Media 가 전송됩니다.

4.2. SDP 협상의 방식

기본적으로 Early Media 가 동작하기 위해서는 위의 그림과 같이 Invite with SDP 로 initial Offer 가 발생되어야 합니다. 위에서 언급한 Media Clipping 은, Invite with SDP (Offer)로 보내는 대부분의 경우 이러한 Media clipping 은 발생하지 않습니다. UAC (송신자)는 200 OK 의 수신 여부와는 상관없이 Offer 를 하자마자 자기 스스로 들어오는 Media 에 대해 재생할 준비를 하도록 되어있기 때문입니다. 그러나, 이렇게 Invite with SDP 만이 있는 것은 아닙니다.

아래 그림에서 보듯이 200 OK 에 밥이 먼저 Offer 를 할 수도 있으며, Update Method 를 이용하여 SDP 를 교환할 수도 있습니다.



4.3. Early Media 의 두 가지 모델

RFC 3960 에 의하면, Early Media 는 다음과 같이 두 가지 모델로 구분할 수 있습니다.

- **Gateway Model**
  이 모델은 SIP 시그널링 상에서 Early Media 에 대한 특별한 언급 없이 동작합니다. 180 Ringing 메시지를 받으면, Local RingBack tone 을 발생시키고, 상대방으로부터 RTP(Early

Media)가 전송되면, Local Ringback tone 발생을 중단하고, RTP 를 재생합니다. 이 Gateway 모델의 문제점은 Forking 및 Security 문제가 있습니다.

- **Application Server Model**
  SIP 시그널시 early media 에 대한 offer/answer 를 가능하게 한다. RFC 3959 에 정의된 Content-Disposition 헤더에 session 또는 early-session 의 새로운 disposition type 을 설정하여 Early-media 가 가능하게 합니다.

여담으로 글을 읽어보면 3960 이 3959 보다 먼저 발표된 글인 듯한데 참 RFC 번호는 반대입니다.

4.4. Ring back tone 재생 정책

PSTN 상에서 수신자가 Alert 메시지를 보내면 Ring back tone 은 PBX 에 의해 재생됩니다. SIP 의 경우 UAS (수신자)에 의해 Ring back 이 재생되지 않으면, UAC 에서 자체적으로 Ring back tone 을 재생하기로 되어 있습니다. 만일, Announcement 또는 special ringing tone 이 UAS 에 재생이 되면, UAC 는 자체적인 재생을 중단하고 UAS 로부터 오는 Media 를 재생하는 것이 일반적입니다. 그러나, UAS 가 early media 전송하려는 의도 없이 Early Offer 를 진행하기도 하고, reliable provisional response 없이 Early media 를 전송하기도 하기 때문에 UAC 는 쉽게 local ring back tone 을 재생해야 할 지 말아야 할지를 결정할 수 없습니다.

local ring back tone 재생에 관련하여 다음과 같은 정책을 구현해야 합니다.

1. **180 Ringing 을 받지 않았다면, local ringing 을 재생하지 않는다**
2. **180 Ringing 을 받았으나 수신되는 Media 패킷이 없다면, local ringing 을 재생한다.**
3. **180 Ringing 을 받았으면서 Media 패킷이 수신된다면, local ringing 을 재생하고 local ringing 을 재생하지 않는다.**

180 Ringing 은 수신 측의 전화기가 울리고 있음을 의미합니다. (Alert)

4.5. Application Server Model 상에서의 Early Media

Early Media session 은 지금까지 이야기한 Regular Session 에서 함께 처리됩니다. 그러나, Application Server Model 은 Early Media 를 위한 별도의 절차를 수행합니다. 따라서, 하나의 호 시그널링에서 Early Media session 과 Regular session 을 동시에 Offer / Answer 를 수행 한 후 전환합니다. 이 때, Content-Disposition Header 에 session 과 early-session 이라고 하여 각각의 쓰임새를 구분합니다.

early session 이 regular session 으로 전환될 때 코덱을 변경할 필요가 없도록 하기 위하여 Early Media session 과 Media session 은 같은 코덱을 사용하는 것을 권장합니다. RFC 3959 의 예제를 보도록 하겠습니다.

앨리스는 밥에게 Content-Disposition: session 헤더를 INVITE 에 추가해서 전송합니다. 이 헤더의 의미는 INVITE 에 포함된 Offer 는 regular session 에 대한 것임을 표시합니다.

```
Content-Type: application/sdp
Content-Disposition: session

v=0
o=alice 2890844730 2890844731 IN IP4 host.sip.com
s=
c=IN IP4 192.0.2.1
t=0 0
m=audio 20000 RTP/AVP 0
```

밥은 183 Session Progress 를 앨리스에게 전송합니다. 이때, 다중 메시지가 포함되었음을 의미하는 Content-Type:multipart/mixed 라는 정보를 함께 보냅니다. 빨간색으로 된 Content-Disposition 헤더를 보시면 session 과 early-session 두 가지가 있습니다. 첫 번째 바운더리인 Content-Disposition:session 은 regular session 에 대한 것으로 기존의 INVITE with offer 에 대한 answer 의 내용입니다. 두 번째 바운더리인 Content-Disposition:early-session 은 Early Media 를 위한 Offer 입니다.

```
Content-Type: multipart/mixed; boundary="boundary1"
Content-Length: 401

--boundary1
Content-Type: application/sdp
Content-Disposition: session

v=0
o=Bob 2890844725 2890844725 IN IP4 host.sip.org
s=
c=IN IP4 192.0.2.2
t=0 0
m=audio 30000 RTP/AVP 0

--boundary1
Content-Type: application/sdp
Content-Disposition: early-session

v=0
o=Bob 2890844714 2890844714 IN IP4 host.sip.org
s=
c=IN IP4 192.0.2.2
t=0 0
m=audio 30002 RTP/AVP 0

--boundary1--
```

앨리스는 INVITE 에 대한 200 OK (수화기를 드는 행동)를 밥으로부터 받기 전까지 Early-Offer 에 대한 Answer 를 할 수 있는 방법이 없습니다. 따라서 이때 PRACK 가 필요합니다. 이 메소드를 통해 아래와 같이 Early Offer 에 대한 Answer 를 수행합니다.

```
Content-Type: application/sdp
Content-Disposition: early-session

v=0
o=alice 2890844717 2890844717 IN IP4 host.sip.com
s=
c=IN IP4 192.0.2.1
t=0 0
m=audio 20002 RTP/AVP 0
```

PRACK 에 대한 200 OK 를 받게 되면, Early Media 를 통해 필요한 대화가 가능합니다. 이미 Regular session 과 Early session 에 대한 Offer / Answer 가 완료되었습니다.

밥이 수화기를 들어 200 OK 를 전송할 때, Early session 은 regular session 으로 전환이 이루어집니다.

Chapter 5

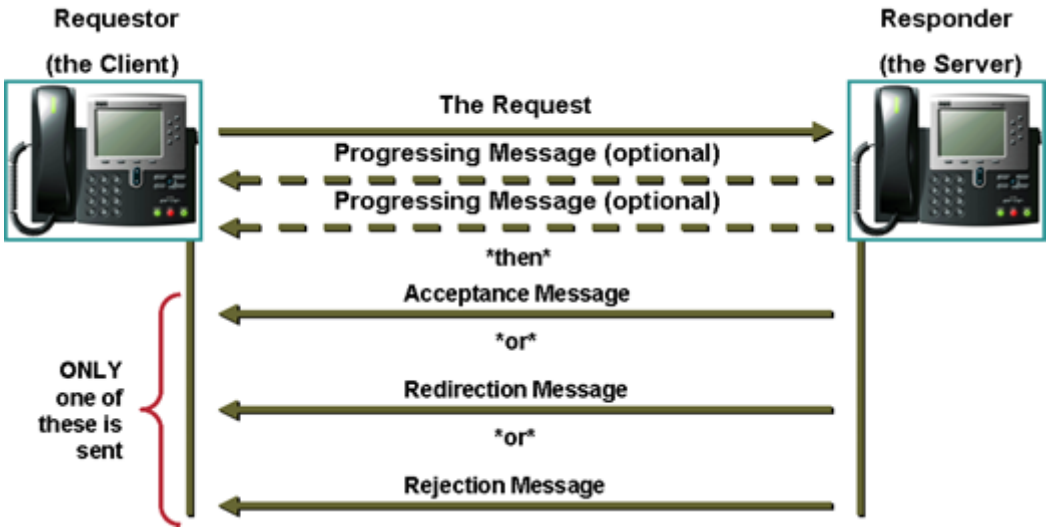
RFC3261 의 주요 메소드(Method)

5.RFC3261 의 주요 메소드

벌써 4 장째입니다. 다른 연재들 처럼 천천히 연재하다가 과중한 업무로 인해 중단되는 일을 반복하지 않기 위해 SIP 연재를 조금 빠르게 진행하겠습니다. 이 연재로 SIP 를 공부하시는 분들에게 도움이 되었으면 합니다. 그럼 시작하겠습니다.

5.1. SIP Methods 개요

이제 기본적인 SIP 의 호 프로시져와 SDP 를 통한 코덱 협상에 대해 충분히 이해하셨을 것입니다. 이제 기본적인 INVITE, 200 OK, ACK 에서 벗어나서 다양한 SIP Method 에 대해 알아보겠습니다. 물론, 200 OK 는 매소드가 아니라 Response 입니다.



위의 그림에서 처럼 SIP 는 Request 에 대한 Response 로 동작합니다. Request 에 대한 응답은 다음과 같은 경우가 있겠습니다.

- **Accept**
  요청을 승인하고, 200 OK 를 송신
- **Reject**
  요청을 거절하고, 사유에 따른 response 를 송신
- **Redirect**
  요청을 수신할 다른 주소를 송신

Response 에 대해서는 6 장에서 자세히 다루도록 하겠습니다.

5.2. SIP 의 주요 매소드

RFC 3261 에 정의된 6 개의 매소드는 다음과 같습니다.

- **INVITE**
  멀티미디어 세션에 참가시키기 위한 서비스 또는 사용자를 초대하기위한 매소드
- **ACK**
  사용자가 INVITE Request 에 최종 응답 (200 OK)를 받았음을 확인하기 위한 매소드
- **BYE**
  기존의 세션을 종료하기 위한 매소드
  멀티미디어 세션의 참가자 누구나 전송 가능
- **CANCEL**
  기존의 Request 를 취소하기 위한 매소드
  만일 기존의 요청에 200 OK 를 받았다면 취소할 수 없음
- **OPTIONS**
  서버의 Capability 를 요청하기 위한 매소드
- **REGISTER**
  User Agent 가 Registrar Server 에 등록하기 위한 매소드

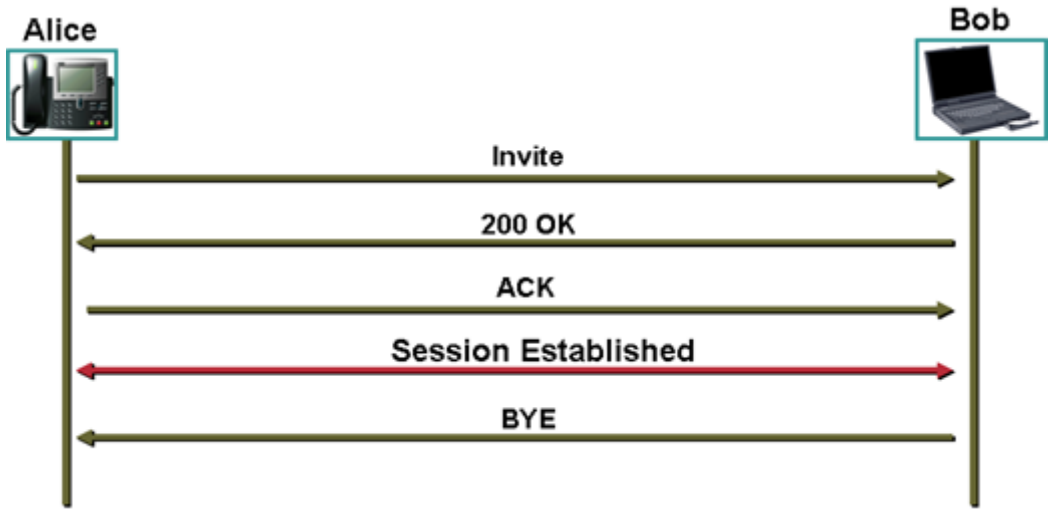
이것 외에 멀티미디어 세션 관리 및 추가 서비스를 위해 다음과 같은 8 개의 매소드가 정의되어 있습니다.

- **INFO (RFC 2976)**
  기존의 성립된 세션 또는 다이얼로그 내에서 추가적인 정보를 전송하기 위한 매소드
- **PRACK (RFC 3262)**
  UAC (User Agent Client)가 임시적으로 Response 를 승인하기위한 매소드
- **SUBSCRIBE (RFC 3265)**
  특정 이벤트를 살펴보기 위해 원격노드에 요청하기 위한 매소드
- **NOTIFY (RFC 3265)**
  특정 이벤트 발생 시 응답하기 위한 매소드
- **UPDATE (RFC 3311)**
  세션 설정 파라미터를 업데이트하기 위한 매소드
- **MESSAGE (RFC 3428)**
  SIP IM (Instant Messaging)을 위한 매소드
- **REFEER (RFC 3515)**
  UA 가 지금 통신 중인 UA 이외의 또 다른 UA 와 통신하기 위한 매소드
- **PUBLISH (RFC 3903)**
  Presence Server 에 UA 의 상태정보를 전송하기 위한 매소드

이렇게 총 14 개의 매소드가 있습니다. 매소드를 알면, SIP 를 통해 어떤 서비스가 가능한 지도 알수 있겠습니다.

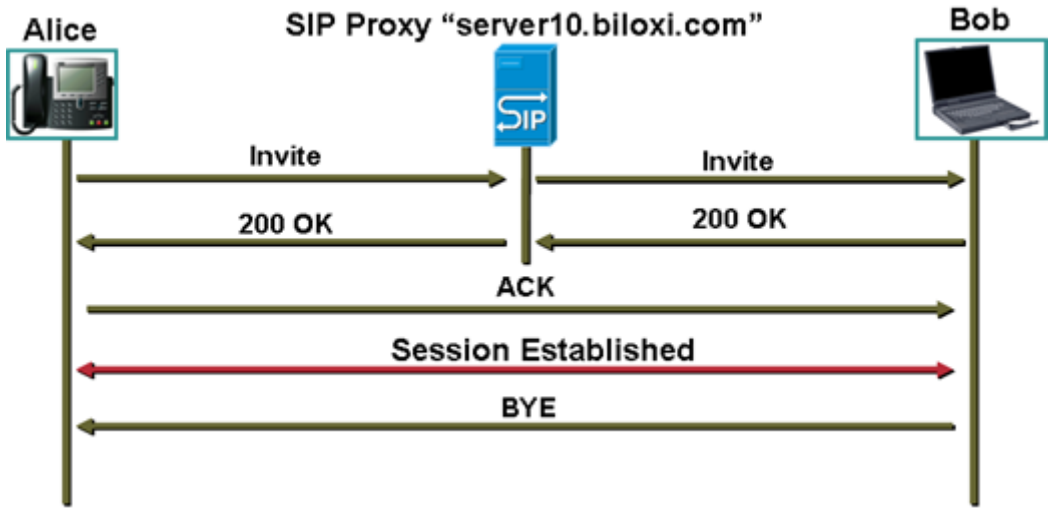
5.3. 일반적인 호 절차 (INVITE, ACK, BYE)

일반적인 SIP 호 절차는 다음과 같습니다. 이 절차에 대한 자세한 설명은 "RFC 3261 분석"에 나와 있으므로 생략하도록 하겠습니다. 아래 그림은 엘리스가 정확하게 Bob 의 주소를 정확하게 알고 있을 경우에 가능할 것입니다.



5.4. 일반적인 호 절차 (INVITE, ACK, BYE with Proxy)

엘리스는 밥의 주소를 알 수 없기 때문에 Proxy 서버를 이용하여 밥의 정확한 주소를 확인할 수 있습니다. 이 때 엘리스는 SIP Proxy 서버의 주소만을 알고 있으면, 모든 목적지로 INVITE 를 전달할 수 있습니다.

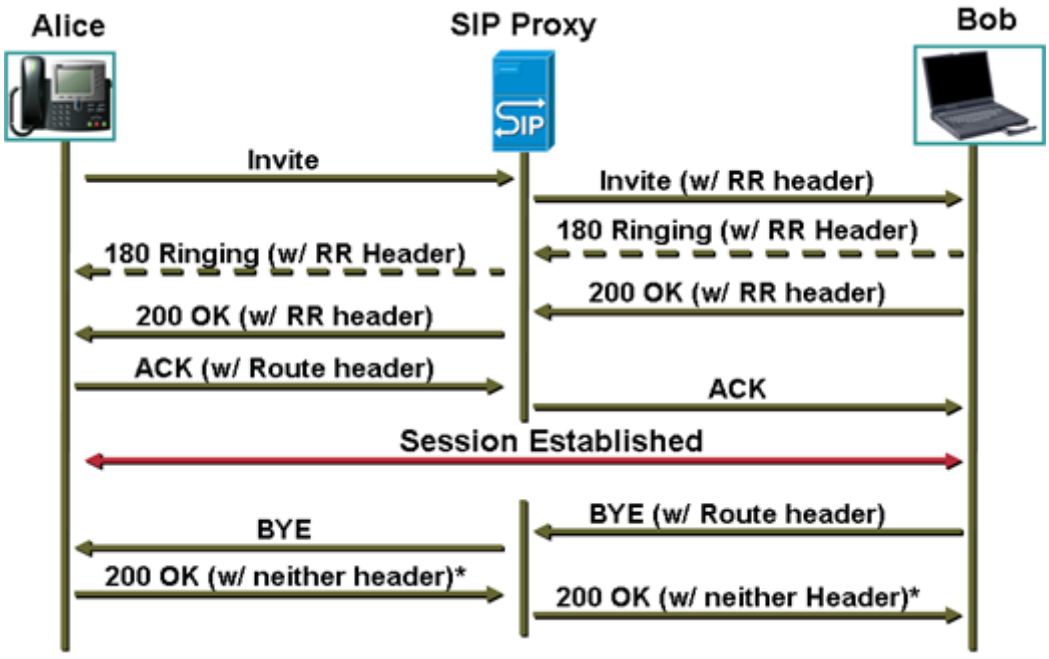


SIP Proxy 는 엘리스로부터 전송된 INVITE request 에 via 헤더를 삽입하여, 200 OK 가 자신을 경유하도록 설정하여 위의 그림과 같은 프로시저가 이루어집니다. 자세한 메시지는 역시 "RFC 3261 분석" 글을 참조하시기 바랍니다.

5.5. Dialog Stateful (INVITE, ACK, BYE with Proxy)

아래 그림와 같이 Invite 뿐만 아니라 ACK 및 BYE 메시지까지 Proxy Server 를 경유하도록 할 필요가 있습니다. 호 다이얼로그 상태를 감시하거나 메시지의 내용을 변경하고자 할 경우, 과금 및 CDR 을 생성해야 할 경우 등에 유용합니다. 이 때 사용하는 것이 Route 헤더 및 Record-Route 입니다. INVITE 에 대한 응답인 200 OK 는 via 필드를 이용하여 메시지로 전송되고, 나머지는 Record-Route 헤더를 이용합니다.

- **Record-Route Header**
  SIP Message 를 확인하려는 모든 SIP Proxy 에 의해 삽입 가능
  SIP Proxy 를 통해 경유되는 dialog (같은 Call-ID)에 대한 Request and Response 에 사용
- **Route Header**
  Dialog 상의 Response 의 Record-Route Header 로 부터 생성



엘리스가 보낸 INVITE 메시지는 SIP Proxy 를 경유하면서, 두 가지 필드가 추가될 것입니다. 하나는 Via 필드이고, 또 하나는 Record-Route 또는 Route 헤더입니다. 200 OK 를 Via 필드를 참조할 것이며, ACK 및 BYE 는 Record-Route 헤더를 참조하여 전송 됩니다.

아래 그림은 엘리스가 보낸 INVITE 메시지가 SIP Proxy 에 의해 변경된 내용을 붉은 색으로 표시한 부분이 추가된 사항이며, 이를 통해 SIP Proxy 가 호 프로시저의 전체에 관여할 수 있게 됩니다.

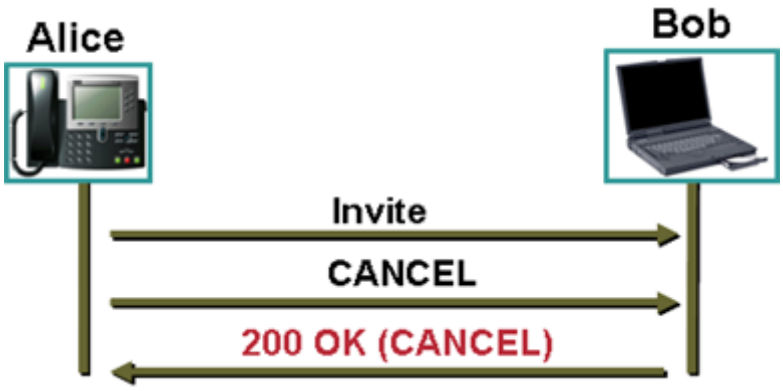
```
INVITE sip:bob@biloxi.com/TCP SIP/2.0
Via: SIP/2.0/TCP server10.biloxi.com
 ;branch=z9hG4bK4b43c2ff8.1
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asdhds ;received=10.1.3.33
Max-Forwards: 69
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Record-Route: server10.biloxi.com
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

(Alice's SDP not shown)
```

5.6. CANCEL

Request 를 취소하고자 할 경우에 사용합니다. 다음과 같은 경우에 CANCEL 메쏘드가 필요할 것입니다.

- 발신자가 전화번호를 누른 후에 링백을 듣다가 바로 수화기를 내려놓는 경우
- Call Forking 과 같은 기능 사용 시 받지 않는 나머지 전화에 대한 INVITE 요청을 취소할 경우
- 상대방이 일정시간 동안 전화를 받지 않는 경우

CANCEL 은 Request 에 대한 취소 요청이므로, Request 에 대한 Response 가 발행되었다면, 취소할 수 없습니다. 예를 들면, INVITE 에 대해 200 OK 를 수신했다면 이미 통화중인 상태가 되는 것입니다. 이때는 BYE 매쏘드를 이용하여 종료해야 할 것입니다.
다음 그림은 CANCEL 의 절차를 나타낸 것입니다. INVITE 에 대한 요청이 수락되기 전에 CANCEL 을 통해 Request 를 취소합니다.

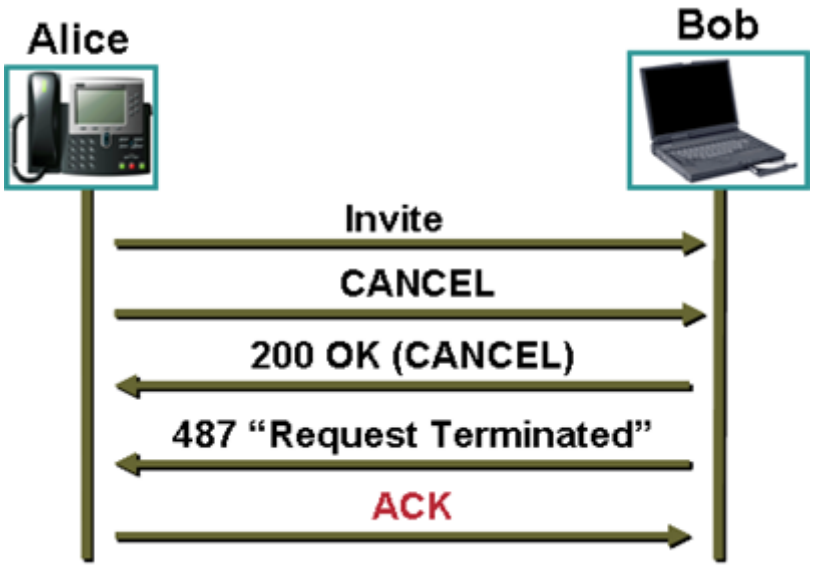


그럼 CANCEL 의 SIP 헤더 내용을 살펴보도록 하겠습니다. 새롭게 추가된 부분만을 살펴보겠습니다.

```
CANCEL sip:bob@192.168.10.20 SIP/2.0
Via: SIP/2.0/TCP 10.1.3.33
 ;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 10197 CANCEL
Contact: <sip:alice@atlanta.com>
Reason: SIP ;cause=486 ;text="Busy Here"
Content-Length: 0
```

Reason 헤더가 새롭게 추가되었으며, CANCEL 의 발행 이유에 대해 명기합니다. 원인은 486 Busy Here 입니다. 즉, INVITE 에 대한 200 OK 를 기다리는 사이에 새로운 호가 인입되어 부득이 기존 호를 종료하게 된 것으로 추정할 수 있습니다.

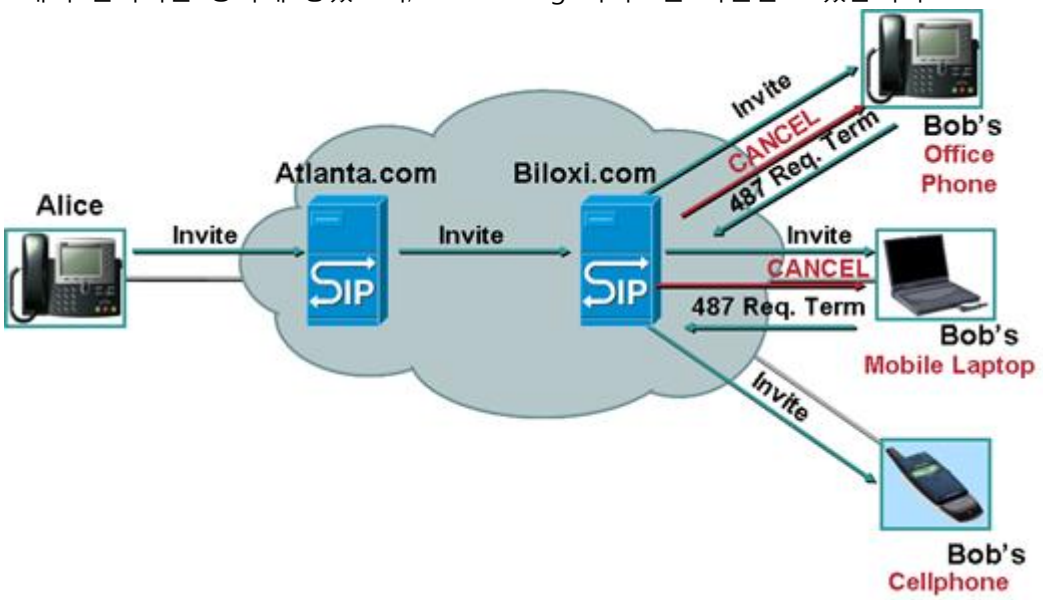
CANCEL 에 대해 밥은 200 OK 로 수락을 하였습니다. 그러나, 최초 Alice 가 전송한 INVITE 에 대한 응답을 밥은 아직 하지 못했습니다. 적당한 Response 는 487 Request Terminated 일 것입니다. 아래 그림은 일련의 호 절차를 확인할 수 있습니다.



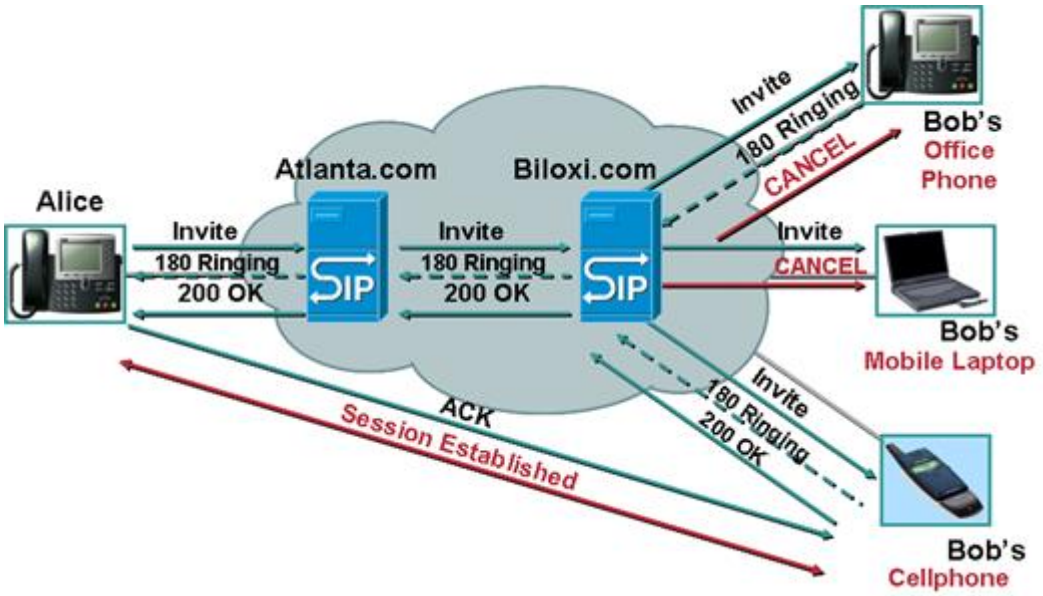
처음으로 INVITE 에 대한 Response 가 200 OK 가 아닌 경우입니다. ^^ 참 4xx Response 에 대한 응답은 항상 ACK 입니다. 4xx 는 4 장에서도 보았듯이 Request 에 대한 실패 (Fail)에 대한 Error 내용입니다.

5.7. Call Forking 에서의 CANCEL

Call Forking 은 아래그림처럼 한 번에 다수의 UA (User Agent)에 INVITE 를 동시 또는 순차적으로 전달하는 것을 의미합니다. 이중에 하나의 UA 가 200 OK 를 전송하면, 나머지 UA 는 CANCEL 을 전송하여 요청을 취소하게 되는 서비스입니다. 아래 그림에서 보면, 밥은 Biloxi.com Proxy 에 3 대의 전화기를 등록해 놓았으며, Call Forking 서비스를 지원받고 있습니다.



위의 그림은 순차적으로 Call Forking 이 진행되는 것이며, 아래는 동시에 Call Forking 이 되는 그림입니다.



이 때 CANCEL 헤더 메시지를 살펴보도록 하겠습니다. 아래그림에서 Reason 헤더의 값은 200 "Call completed elsewhere" 입니다.

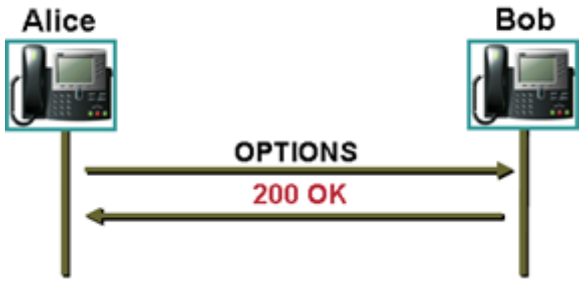
```
CANCEL sip:bob@192.168.10.20/TCP SIP/2.0
Via: SIP/2.0/TCP server10.biloxi.com
 ;branch=z9hG4bK4b43c2ff8.1
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Server10 <sip:server10.biloxi.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 6187 CANCEL
Contact: <sip:server10.biloxi.com>
Reason: SIP ;cause=200 ;text="call completed elsewhere"
Content-Length: 0
```

5.8. OPTION 의 개요

UA 는 OPTION 메쏘드를 통해 UA 또는 SIP Proxy 의 Capability 를 링을 울리지 않고도 조용히 확인할 수 있습니다. 확인할 수 있는 정보는 다음과 같습니다.

- Method
- Content types
- Extensions
- Codecs

아래 그림은 간단한 OPTION 요청과 응답을 나타낸 것입니다.



OPTION 의 내용을 살펴보겠습니다. Allow 와 Accept 헤더를 통해 엘리스는 밥에게 정보를 제공합니다.

```
OPTIONS sip:bob@192.168.10.20 SIP/2.0
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK77i832k9
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e6Kr456@pc33.atlanta.com
CSeq: 22756 OPTIONS
Contact: <sip:alice@pc33.atlanta.com>
Allow: INVITE, ACK, OPTIONS, BYE, CANCEL, REFER,
  SUBSCRIBE, NOTIFY, MESSAGE, UPDATE
Accept: application/sdp, application/pidf-xml
Content-Length: 0
```

아래 그림에서 보듯이 밥은 200 OK 를 통해 다음과 같은 정보를 제공합니다.

- 잘 알려진 Contact (Contact 헤더)
- 지원 메쏘드 (Allow 헤더)
- 언어 (Accept-language 헤더)
- Message Body type (Accept 헤더)
- 지원 코덱 (SDP) : 포토번호가 0 으로 설정됨

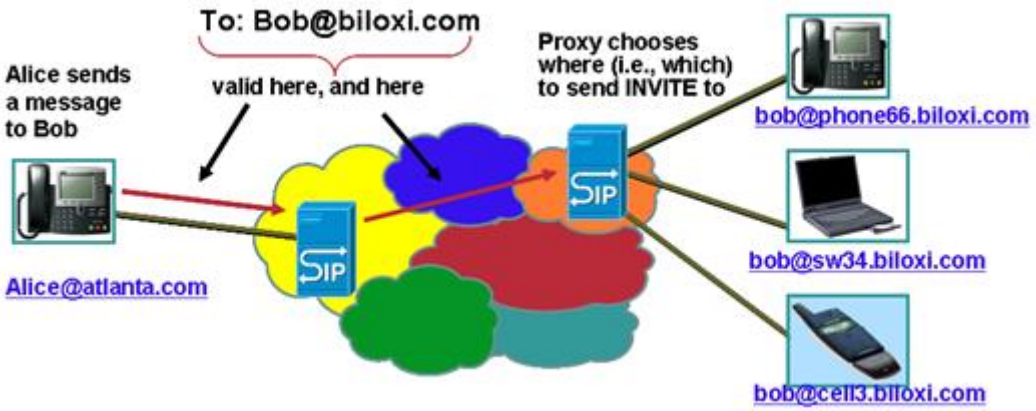
```
SIP/2.0 200 OK
Via: SIP/2.0/TCP sip:alice@atlanta.com
 ;branch=z9hG4bK77i832k9
To: Bob <sip:bob@biloxi.com>; tag=a6c85e3
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e6Kr456@pc33.atlanta.com
CSeq: 22756 OPTIONS
Contact: <sip:bob@biloxi.com>
Contact: <sip:bob_home@biloxi.com>
Allow: INVITE, ACK, OPTIONS, BYE, CANCEL, REFER,
  NOTIFY, MESSAGE
Accept: application/sdp, text/plain, image/jpeg
Accept-language: en, fr
Content-Type: application/sdp
Content-Length: 274

(Bob's SDP indicating codecs supported)
```

5.9. REGISTER

SIP 에 있어서 등록과정은 필수적인 요소입니다. SIP Proxy Server 는 옵션이지만, 단말의 숫자가 증가할 경우 모든 경로에 대한 정보를 단말이 보유하는 것은 불가능하므로 일반적으로 SIP Proxy Server (일반적으로, Registra Server 와 함께 구현됨)를 사용하며, 단말들은 등록과정을 진행합니다. 등록과정을 통해 SIP Proxy Server 는 단말의 현재 위치를 확인할 수 있습니다. 즉, address-of-record (AOR) URI 와 Contact address 를 바인딩하는 것입니다. 여기서 AOR 과 Contact address 에 대해 짚어보도록 하겠습니다.

- AOR (Address of Record)
  외부 도메인에서 현재 통신하려는 사용자
  Bob@biloxi.com
- Contact Address
  등록된 단말의 특정 주소, 즉, 사용자가 나중에 통신하려는등록된 단말의 특정 주소
  bob@phone66.biloxi.com

어렵습니다. 아래 그림을 보면 좀 더 쉽게 이해할 수 있습니다.



엘리스가 밥과 통화를 할려고 합니다.그러면 엘리스는 밥에게 접근하기 위해 INVITE 를 보낼 것입니다. 밥은 3 개의 단말을 가지고 있으며, 어느것이 현재 등록되어 있는 지는 biloxi.com 의 SIP Proxy Server 만이 알고 있습니다. 따라서, biloxi.com 도메인 밖에서는 Bob@biloxi.com 이 가장 유효한 주소이며, biloxi.com 의 SIP Proxy Server 에서는 등록된 단말을 찾을 수 있는 주소가 유효해 지는 것입니다. 이런 AOR 과 Contact Address 를 바인딩하는 것이 등록 (Registration)입니다.

아래 그림은 일반적인 등록 과정입니다. 단순히 REGISTER 요청에 200 OK 로 응답하면, 등록이 되지만, 자세한 메시지에 대해서도 짚고 넘어가도록 하겠습니다.

엘리스는 자신의 정보를 SIP Proxy Server 에 등록을 시도합니다. 이때, Proxy Server 의 주소를 아는 것은 다양한 방법이 있겠지만,일반적으로 사전 설정을 통해 UA 가 등록을 시도할 수 있도록 합니다. 이때 메시지는 아래와 같습니다. 엘리스는 21600 초 동안 등록을 유지해 줄 것을 Expires 헤더를 통해 알립니다. 200 OK 응답의 경우 3600 초 동안 등록 정보를 유지하겠다고 합니다.

```
REGISTER sip:server19.atlanta.com SIP/2.0
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bk2l55n1
To: Alice <sip:alice@atlanta.com>
From: Alice <sip:alice@atlanta.com>;tag=283074
Call-ID: a84b4g96te10@pc33.atlanta.com
CSeq: 31862 REGISTER
Contact: <sip:alice@10.1.3.33>
Expires: 21600
Content-Length: 0
SIP/2.0 200 OK
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bk2l55n1; received=10.1.3.33
To: Alice <sip:alice@atlanta.com>; tag=a6c85e3
From: Alice <sip:alice@atlanta.com>;tag=283074
Call-ID: a84b4g96te10@pc33.atlanta.com
CSeq: 31862 REGISTER
Contact: <sip:alice@pc33.atlanta.com>
Contact: <sip:alice@cm9013.atlanta.com>
Service-Route: <sip:bigbox3.atlanta.com;lr>
Expires: 3600
Contact-Length: 0
```

200 OK 의 Contact 에는 항상 현재 바인딩되어 있는 주소 정보를 표시합니다.
또한, 200 OK 에는 Service-Route 헤더가 있습니다. 이 헤더는 RFC 3608 SIP Extension Header Field for Service Route Discovery During Registration 에 정의되었습니다. 이 헤더는 등록과정에서 UA 가 SIP Home Proxy Server 정보를 획득할 수 있도록 해 주는 것입니다.

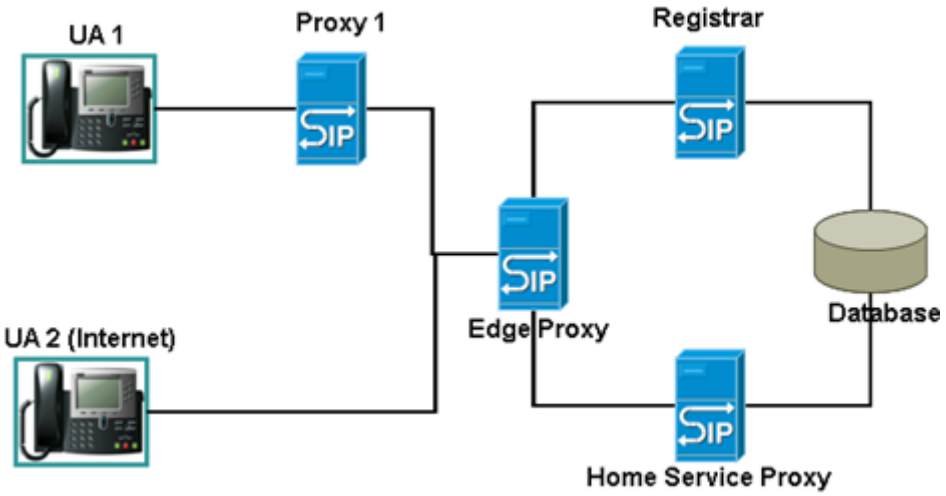
5.10. RFC 3608 상의 Service Route Header 의 이해

등록 과정에서 UA 의 AOR 과 Contact address 가 바인딩 되어 있으므로, SIP Proxy 로 온 INVITE 요청은 Contact address 로 정확히 전달될 것입니다. 반대로, UA 에서 Outbound 를 위해 INVITE 를 생성하여 전송하려고 할 때 사용할 수 있는 Proxy 에 대한 정보를 획득하는 방법이 필요합니다. 다음과 같이 여러가지 방법이 있습니다.

- UA 에 수동 설정
  UA 설정 정보 관리와 업데이트에 대한 이슈가 발생하지만, 가장 많이 사용합니다. Proxy 가 자주 바뀐다고 한다면, 정말 골치가 아픈 일이 발생하게 될 것입니다.
- HTTP 또는 TFTP 와 같은 프로토콜을 이용하여 UA 의 설정 정보 전달
  이것 또한 많이 사용하는 방법입니다. 방화벽이 많으면 관리가 복잡해지는 문제가 있고, UA 는 SIP 외에 다른 프로토콜 엔진을 구동해야 할 것입니다.
- Lookup table 활용
  데이타베이스에 대한 오버헤드 발생

위의 언급된 것 말고도 RFC 3608 에는 302 Temprary Moved Response 를 활용하는 방법이나 Record-Route 헤더를 REGISTER 매쏘드에서도 활용하는 방법등이 열거되어 있습니다만 결국 RFC 3608 의 Service Route Extension Header 를 사용할 것을 권고합니다. 실제 Enterprise 급 장비는 이런 필드가 필요없을 수도 있습니다만, 거대 3GPP 망 같은 경우에는 상당히 다른 경우가 될 것입니다.

다음 그림과 같은 망 구조로 된 3GPP 망(핸드폰 망) 을 생각해 봅시다.



UA1 은 Visited Network 에 있으며, UA1 은 수동 설정 또는 DHCP 에 의해 Proxy 1 의 주소를 획득한다고 가정합니다. 여기서 UA1 실제 Outbound Service 를 Home Service Proxy 의 주소를 획득하는 것이 필요합니다. 이 때 REGISTER request 에 대한 200 OK 응답에 Service Route 헤더를 이용하여 알려줍니다. 이과정은 RFC 3608 에 자세히 나와 있습니다.

UA1 이 다음과 같은 REGISTER 메시지를 전달하면, Proxy 1 은 Response 를 받기 위해 Via 헤더를 추가하여 Edge Proxy 로 전달할 것입니다. Edge Proxy Server 도 Via 헤더를 추가하여 Registrar Server 에 전달합니다. 즉 REGISTER 요청에 3 개 Via 헤더가 추가될 것입니다.

```
REGISTER sip:HOME.EXAMPLE.COM SIP/2.0
Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKcR1ntRAp
To: Lawyer <sip:UA1@HOME.EXAMPLE.COM>
From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=981211
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
...
```

Registrar Server 는 SIP URI (AOR) 주소와 Contact 주소를 바인딩합니다. 또한, Registrar Server 는 수동 설정 또는 기타 다른 방식을 통해 Home Service Proxy 및 Edge Proxy 의 주소를 알고 있으므로, 이를 통해 Service-Route 를 생성합니다. 다시 Via 경로를 따라 최종 UA1 에 전달된 200 OK 에는 다음과 같은 정보를 담게 됩니다.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKcR1ntRAp
To: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=87654
From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=981211
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
Service-Route: <sip:P2.HOME.EXAMPLE.COM;lr>,
        <sip:HSP.HOME.EXAMPLE.COM;lr>
...
```

Service-Route 헤더에는 Edge Proxy 와 Home Service Proxy 의 정보가 담기게 됩니다. UA 1 은 Service Route 정보를 저장하여 향후 발생할 요청에 사용합니다. 이렇게 간단하게 Proxy Server 의 정보를 획득할 수 있습니다. 이를 통해 UA2 를 찾아가는 INVITE 메시지에는 Route 헤더를 포함하여 전달되게 됩니다.

```
INVITE sip:UA2@HOME.EXAMPLE.COM SIP/2.0
Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKnashds7
To: Customer <sip:UA2@HOME.EXAMPLE.COM>
From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=456248
Call-ID: 38615183343@s1i1l2j6u
CSeq: 18 INVITE
Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
Route: <sip:P2.HOME.EXAMPLE.COM;lr>,
    <sip:HSP.HOME.EXAMPLE.COM;lr>
...
```

실제 INVITE 메시지는 Proxy 1 --> Edge Proxy --> Home Service Proxy 를 거쳐 전달됩니다. Home Service Proxy 는 UA2 의 Contact Address 를 획득하여 INVITE 요청을 Edge Proxy 로 전달하고, Edge Proxy 는 UA2 로 직접 전달하게 됩니다. 아래 그림은 Home Service Proxy 가 Edge Proxy 로 전달한 INVITE 메시지 입니다.

```
INVITE sip:UA2@UAADDR2.HOME.EXAMPLE.COM SIP/2.0
Via: SIP/2.0/USP HSP.HOME.EXAMPLE.COM:5060;branch=z9hG4bKHSP10120323
Via: SIP/2.0/UDP P2.HOME.EXAMPLE.COM:5060;branch=z9hG4bKiokioukju908
Via: SIP/2.0/UDP P1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bK34ghi7ab04
Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKnashds7
To: Customer <sip:UA2@HOME.EXAMPLE.COM>
From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=456248
Call-ID: 38615183343@s1i1l2j6u
CSeq: 18 INVITE
Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
Record-Route: <sip:HSP.HOME.EXAMPLE.COM;lr>
Record-Route: <sip:P2.HOME.EXAMPLE.COM;lr>
Record-Route: <sip:P1.VISITED.EXAMPLE.ORG;lr>
...
```

Chapter 6

RFC3261 의 Response 이해

6.1. RFC 3261 의 Response

이제 절반 정도 왔습니다. 6 장 이후의 차례가 아무래도 대폭 변경될 듯합니다. 생각보다 내용이 점점 많아지기도 하고, 불필요한 부분이 생기기도 합니다. 차례가 바뀌거나 없어져도 놀라지 마시고, 읽어 주시기 바랍니다. SIP 에 대한 전체적인 내용이 담기도록 노력하겠습니다.

개요

SIP 는 Request and Response 프로토콜이라고 설명 드렸습니다. 지금까지는 Request 위주로 살펴보았다면, 이 장에서는 Response 에 대해 살펴보겠습니다. 아래 표는 자주 사용되는 Response 입니다.

	Description	Examples
1xx	Informational – Request received, continuing to process request.	100 Trying 180 Ringing 181 Call is Being Forwarded 183 Session Progressing
2xx	Success – Action was successfully received, understood and accepted.	200 OK 202 Acceptable
3xx	Redirection – Another SIP Element needs to be contacted in order to complete the request.	300 Multiple Choices 301 Moved Permanently 302 Moved Temporarily
4xx	Client Error – Request contains bad syntax or cannot be fulfilled at this server.	401 Unauthorized 406 Not Acceptable 407 Proxy Authentication Required 486 Busy Here 487 Request Terminated 488 Not Acceptable Here
5xx	Server Error – Server failed to fulfill an apparently valid request.	502 Bad Gateway 503 Service Unavailable
6xx	Global Failure – Request is invalid at any server.	600 Busy Everywhere 603 Decline

6.1.1. 1XX Informational Responses

Informational Responses 는 Request 에 대해 Final Response 전에 약 200ms 이상의 시간이 소요되면 서버가 처리중임을 위해 전송됩니다. 이 Response 에는 SDP 와 같은 Body 가 함께 전송될 수 있습니다.

- 100 Trying
  Request 는 다음 서버로 전송되어 처리 중임을 의미합니다. 만일 데이터베이스에 대한 쿼리가 진행 중이라면, 응답이 늦어져 최초 Request 가 재송신될 수 도 있습니다. 이를 막기 위해 100 Trying 을 UAC 로 전송합니다.
- 180 Ringing
  가장 많이 보는 Response 가운데 하나입니다. UAS 가 사용자에게 Alerting 중임을 나타내고 이를 수신한 UAC 는 Local Ringback tone 을 재생할 것입니다.
- 181 Call Is Being Forwarded
  만일 UAS 가 Call Forward Busy 또는 Call Forward No answer 로 착신 전환되어 있을 경우 Proxy 는 INVITE 를 착신전환 번호로 재송신 중임을 UAC 에게 알려주기 위해 사용합니다.

- 182 Queued
  착신 측 UAS 가 일시적으로 통화를 할 수 없는 상태일 때 호를 거절하지 않고 큐에 넣어 놓았다가 UAS 가 통화가 가능해지면 호를 재 연결합니다. 이런 경우는 IPCC 와 같은 곳에서 많이 유용할 것입니다. 모든 상담원이 통화중이라면, 큐에 넣어 놓았다가 다시 상담 대기상태로 전환된 상담원에게 연결하는 상황에 사용됩니다.
- 183 Session Progress
  위에 열거되지 않은 상황에 대한 호 진행 정보를 전송할 때 사용한다고 보시면 됩니다.

6.1.2. 2XX (Successful)

다들 아시는 것일 것입니다. Request 가 정상적으로 처리되었음을 의미합니다. 대표적인 것이 200 OK 입니다.

6.1.3. 3XX Redirect

사용자의 새로운 UA 에 대한 정보로 응답합니다.

- 300 Multiple Choices
  사용자가 여러 개의 단말을 소유할 수 있으며, 이에 따라 선호하는 UA 로 호를 진행합니다.
- 301 Moved Permanently
  사용자가 더이상 Request-URI 의 주소로 찾을 수 없음을 의미하며, reponse 에 포함된 Contact Header 주소로 re-INVITE 를 진행해야 합니다. 또한, Local Directories, 주소록 등에 정보를 업데이트 해야 합니다.
- 302 Moved temporarily
  사용자가 다른 곳으로 옮겼으며, reponse 에 포함된 Contact Header 주소로 re-INVITE 를 진행해야 합니다. 301 과 차이라면, 일시적인 이동임으로 Local Directories 에 업데이트할 필요가 없습니다.
- 305 Use Proxy
  UAS 에 도달한 Request 가 Proxy 를 경유하지 않아 경유하도록 Contact address 를 보냅니다.
- 380 Alternative Service
  이 Reponse 는 한 번도 보지 못한 것입니다. 진행된 호는 실패했지만, 다른 서비스가 가능함을 의미한다고 되어 있는 데 무슨 말인지 모르겠습니다. ^^
-

6.1.4. 4XX Request Failure

호를 진행할 수 없는 사유가 나타납니다. UA 는 메시지 변경 없이 같은 Request 를 반복해서는 안됩니다. 4XX 는 매우 많은 Reponse 가 정의되어 있습니다. 자세한 사항은 RFC 3261 21 장을 보시기 바랍니다. 저도 적다가 포기했습니다. -,-:?

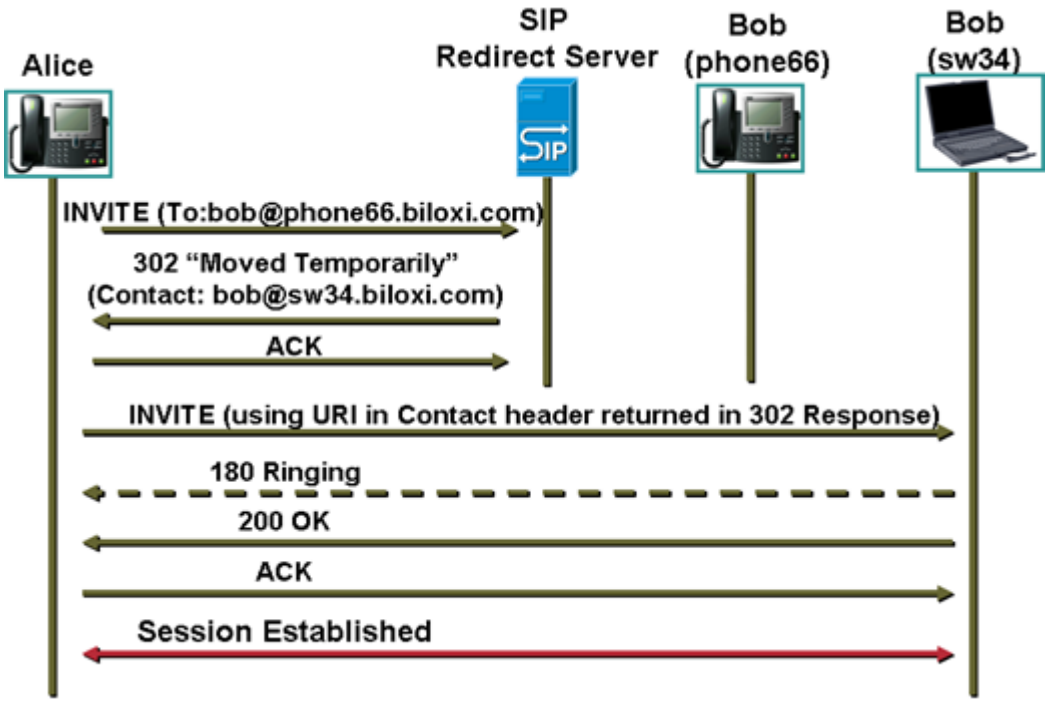
- 400 Bad Request
  잘못된 문구나 메시지 포맷을 따르고 있지 않아 처리될 수 없음을 의미합니다. 예를 들면, 필수적인 SIP 헤더가 빠져있다던가 할 수 있을 것입니다.
- 401 Unauthorised & 407 Proxy Authentication Required
  Request 는 사용자 인증을 요구합니다. Registrar 또는 UAS 에 의해 401 Response 가 발행되며, Proxy Server 에 의해 407 Response 가 발생됩니다.
- 403 Forbidden
  서버는 Request 를 이해했지만, 처리를 거절합니다.
- 404 Not Found
  Request-URI 에 있는 도메인 주소가 존재하지 않음을 의미합니다.
- 406 Not Acceptable
  Accept Header 에 열거되지 않은 사항을 요구할 경우 전송됩니다.
- 408 Request Timeout
  일정 시간안에 Request 에 대한 응답을 할 수 없음을 의미합니다.
-

6.1.5. 5XX Not Implemeted

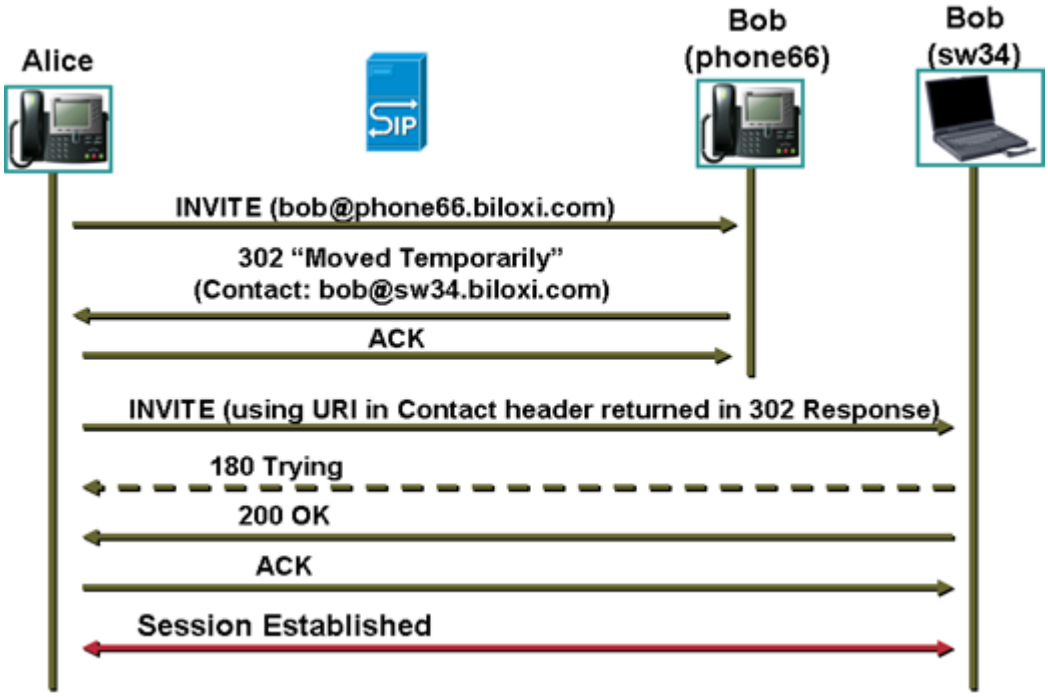
서버 상에 구현되지 않은 서비스를 요구할 경우에 사용됩니다. 자세한 사항은 생략합니다.

6.1.5. Redirect (302 "Moved Temporarily")

다음 그림은 엘리스가 밥의 데스크탑 전화기로 INVITE 요청을 한 경우입니다. 밥은 Call Forward All (착신 전환) 또는 착신 선호 UA 변경과 같은 설정을 한 상태로 가정할 수 있습니다. 이때 INVITE 요청에 대한 302 "Moved Temporarily" Response 가 Redirect Server 에 의해 전송됩니다. 이 때 302 Response 내의 Contact 헤더는 re-INVITE 를 전송할 주소가 명기되고 re-INVITE 가 밥의 소프트폰으로 전송됩니다.



일반적으로 Redirect Server 는 SIP Proxy 와 함께 구현이 됩니다. 만일 Call Forward All 을 설정하였으나, SIP Proxy 내의 Bob 의 프로파일에 따로 설정변경이 이루어지지 않고 전화기 상에서만 구현되었다면 아래 그림과 같은 Redirect 가 발생할 것입니다.

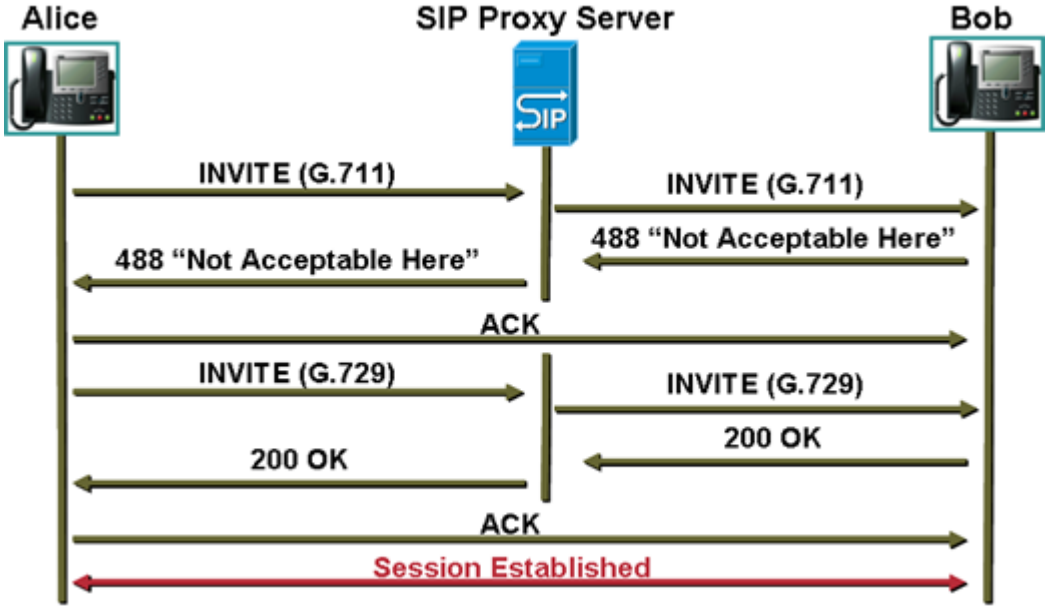


실제 호 절차는 SIP Proxy 에 의해 이루어지나 Bob 의 전화기에 의해 발생하나 마찬가지 절차를 수행한다는 것을 알수 있습니다.

착신측이 지원하지 않는 코덱을 사용할 경우 (488 "Not Acceptable Here")

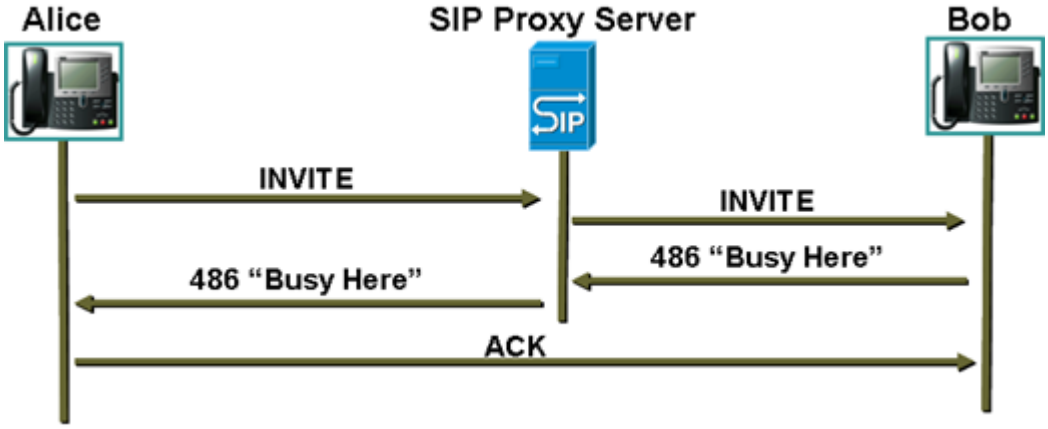
엘리스는 G.711 로 호를 개통하기 위해 INVITE 요청을 밥에게 보내지만, 밥은 G.729 만을

지원합니다. 이 때 488 "Not Acceptable Here"라고 응답을 하며, 이 메시지에 선호되는 코덱이 SDP 메시지에 명기될 것입니다. 따라서, re-INVITE 에 G.729 로 개통하자는 호가 진행됩니다.



통화중 (Called party Busy, 486 Busy Here)

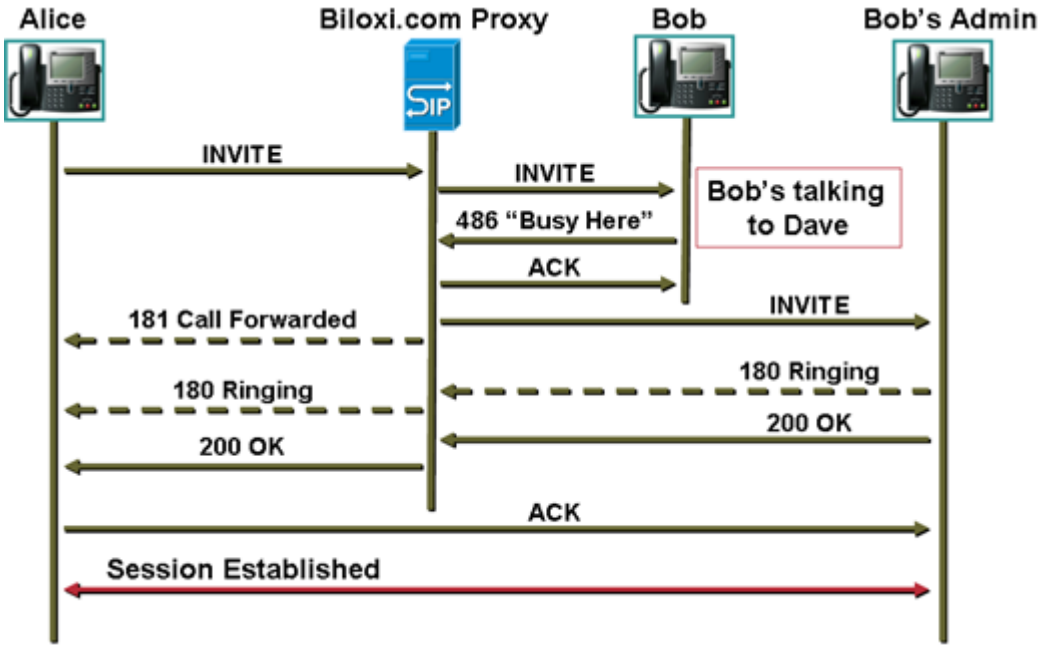
SIP Proxy 가 밥의 상태 정보를 인지하지 못할 경우 INVITE 요청이 밥에게 전달되고, 밥은 다른 사람과 통화중이라면 "486 Busy Here" 에러 메시지를 전달합니다.



여기에서는 통화중일 경우를 가정하여 설명했지만, UAS (착신측 UA)가 호를 진행할 수 없는 모든 상황에 대해 보낼 수 있습니다. 또한, 비슷한 Response 로는 600 Busy Everywhere 와 488 Not Acceptable Here 도 가능합니다. 600 은 호를 응답할 수 있는 다른 시스템도 없을 경우이며, 호를 거절 하기할 경우 488 응답을 전달하며, 정확한 거절사유가 명기되어야 합니다.

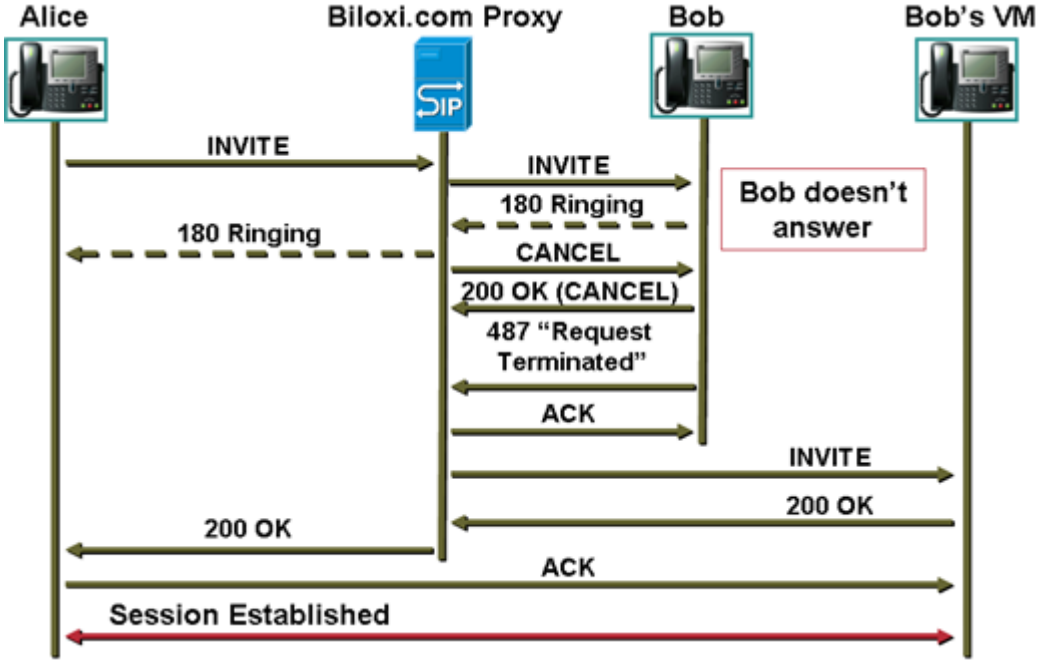
착신전환

다양한 경우나 상황에서 착신전환은 이루어질 수 있습니다. Call Forward All, No answer, Busy 이외에도 Proxy 의 설정에 따라 다양한 방식의 착신 전환을 지원합니다. 각각의 상황에 대해 살펴보겠습니다. 아래 그림은 Call Forward Busy 상황입니다.



통화중일 경우의 호 절차는 이미 살펴보았으며, 486 "Busy Here"를 받은 Proxy 는 INVITE 를 다시 정해진 Call Forward Busy 번호로 요청하면서 동시에 발신자인 엘리스에게 181 "Call is being Forwarded"를 전송합니다. 밥의 관리자로 포워딩 된 호는 Ringing 메시지를 전달받아 엘리스에게 전달하게 되어 나머지 절차가 진행됩니다.

그렇다면 Call Forward No Answer 의 경우는 어떻게 될지 아래 그림을 살펴보겠습니다.

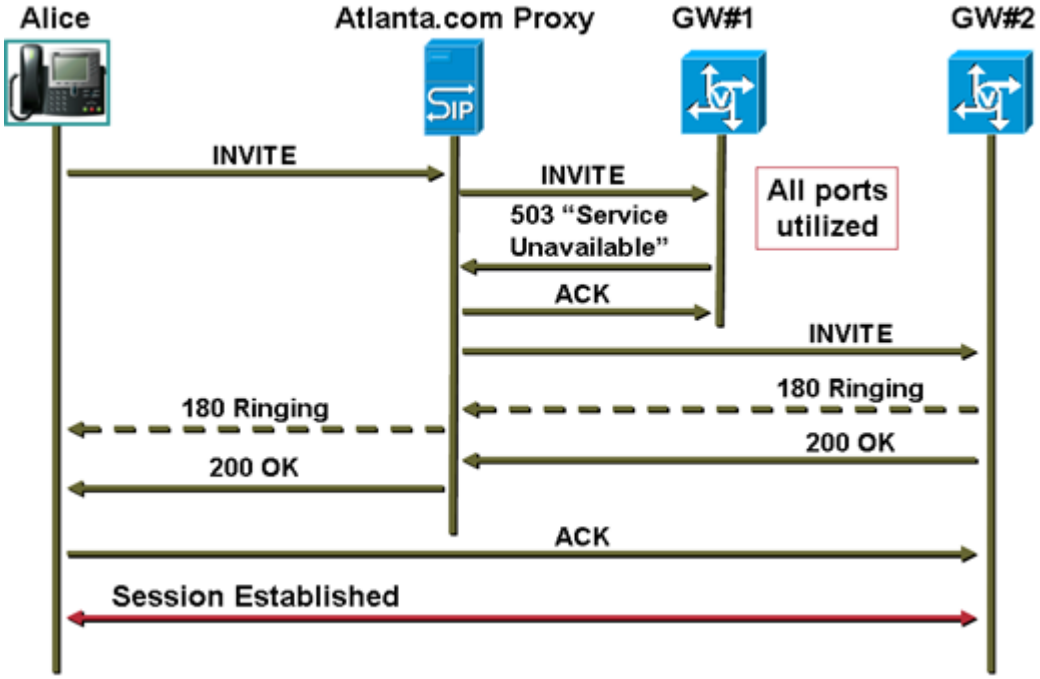


180 Ringing 후에도 일정시간동안 200 OK 가 Proxy 로 전달되지 않으므로 Proxy 는 CANCEL 을 요청하고, Call Forward No Answer 번호로 호전환이 이루어집니다. 착신측이 음성사서함이므로

별도의 180 Ringing 메시지 없이 바로 200 OK 가 전달될 것입니다. 또한 바로 음성 프롬프트가 진행될 것입니다.

6.1.6. Gateway Congestion

만일 Proxy 가 보낸 INVITE 를 처리할 DSP 가 Gateway 내에 없다면, 아래의 그림과 같은 503 "Service Unavailable"로 응답합니다.이에 Proxy 는 두 번째 게이트웨이로 다시 호를 시도합니다. 이런 프로세스가 진행될 수 있도록 Proxy 에 설정되어 있어야 합니다.

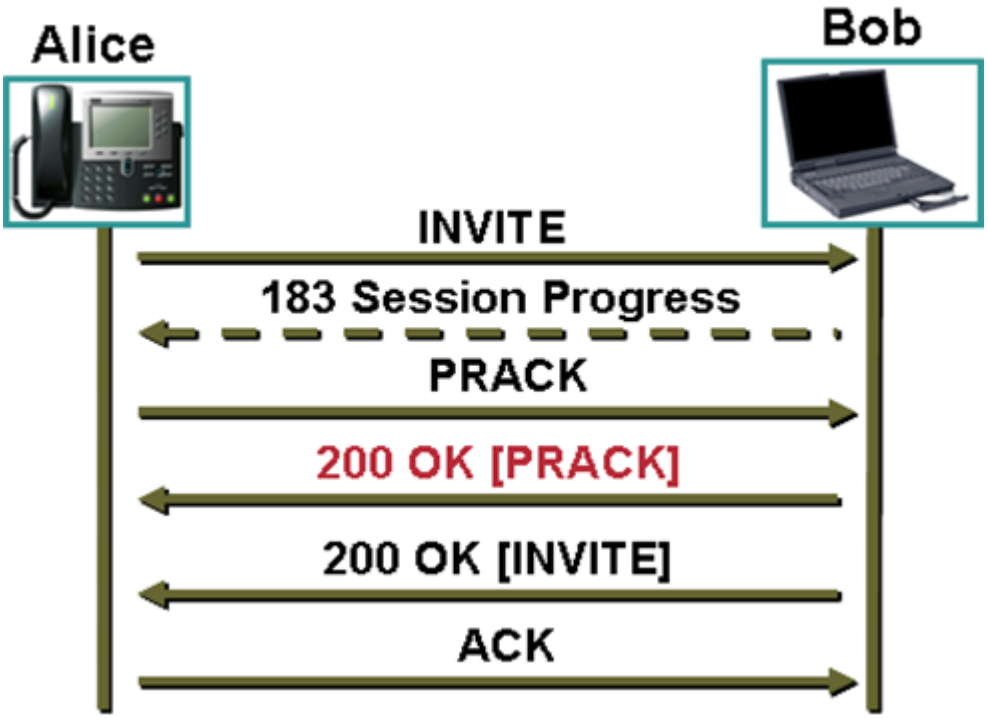


Chapter 7

RFC3261 의 PRACK 이해

7.1. PRACK 개요

PRACK 는 Provisional Response ACKnowledgement 의 약어로써, 아직 설립되지 않은 세션에 대한 신뢰할 수 있는 Provisional ACK 를 제공하는 것입니다. 이미 제 3 장 Early Media in SDP 에서 잠시 다루었던 메쏘드입니다. UAC 가 INVITE Request 를 보내면, UAS 는 100 Trying 또는 183 Session Progress 와 같은 메시지를 200 OK Response 이전에 보내므로 여기에 필요한 정보를 실어 보냅니다. 그러나, UAC 의 입장에서는 유일하게 INVITE 에 대한 200 OK 를 받으면, ACK 를 통해 응답이 가능합니다. 즉, 200 OK 이전에 신뢰할 수 있는 응답을 제공하기 위한 방안이 없습니다. 따라서, PRACK 을 통해 응답을 수행하게 됩니다.



잘 이해가 되지 않을 것입니다. 다시 한번 위의 그림을 보면서 설명 드리겠습니다. PRACK 은 INVITE 에 대한 200 OK Final Response 전에 UAC 에 의해 생성되는 것이며, 183 Session Progress 라는 Provisional Response 에 대한 응답이 PRACK 에 포함되는 것입니다. 따라서, Provisional Response 가 신뢰할 수 있도록 되는 것입니다. PRACK 이 없다면, 183 Session Progress 에 대해 UAC 는 신뢰할 수 있는 응답을 제공할 수 없습니다.

즉, RFC 3261 상의 Response 는 Final 과 Provisional 두 가지 타입으로 정의합니다. Final Response 는 Request 에 대한 처리 결과로써 생성되어 신뢰할 수 있도록 전송됩니다. 예를 들면, INVITE 에 대한 200 OK 가 주어지는 것이며, 또한 200 OK 에 대한 ACK 가 주어지는 것입니다. Provisional Response 는 Request 에 대한 처리 중에 정보를 제공하기 위해 생성되지만, 신뢰할 수 있는 방안을 제공하지 못합니다. 이 Provisional Response 에 대한 신뢰할 수 있는 전송 방안을 제공하는 것이 PRACK 입니다. PRACK 는 일반적인 Request 와 똑같이 동작하여 200 OK Response 를 받습니다.

PARCK 은 INVITE 에 대한 100 Trying 이외의 101 부터 199 Response 에 대해서만 제공됩니다. 사실 100 Trying 은 hop-by-hop 으로 이루어지는 것으로 end-to-end 메커니즘이 아니기 때문입니다. 밥과 엘리스 사이에 SIP Proxy 가 2 개가 있다고 생각해보시길 바랍니다. 100 Trying 은 밥으로부터 전달되는 것이 아니라, INVITE 를 받은 SIP Proxy 에 의해 생성됩니다. 지금까지 주욱 연재를 읽어오신 분들에겐 당연한 얘기입니다.

7.2. PRACK 관련 헤더 분석

위의 그림의 호 절차대로 각 SIP 헤더를 살펴보도록 하겠습니다. 아래 그림처럼 INVITE with SDP 를 전송할 때, UAC 는 Requires 헤더에 100rel 메시지를 포함합니다. 100rel 은 Provisional Response 에 대한 신뢰성을 제공하기 위한 Option Tag 로써, UA 는 신뢰할 수 있는 Provisional Response 주고 받을 수 있음을 의미합니다.

```
INVITE sip:bob@192.168.10.20 SIP/2.0
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Requires: 100rel
Content-Type: application/sdp
Content-Length: 142

(Alice's SDP not shown)
```

아래 그림은 183 Session Progress 로써 Provisional Response 입니다. 각 Provisional Response 에는 Rseq 헤더를 통하여 sequence number 를 제공합니다. 이때 UAS 가 100rel 을 지원하지 않는다면, 420 Bad Extension 을 통해 거절하며, Unsupported 헤더에 사유를 명기합니다.

```
SIP/2.0 183 Session Progress
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asdhds
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
RSeq: 813520
Contact: <sip:alice@pc33.atlanta.com>
Content-Type:  application/sdp
Content-Length: 235

(Bob's (different) SDP not shown)
```

아래 그림은 PRACK SIP 헤더입니다. Rack 헤더를 통해 Rseq 의 sequence number 를 포함하여 ACK 함을 나타냅니다.

```
PRACK sip:bob@192.168.10.20 SIP/2.0
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asi98JK
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159
RAck: 813520 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Length: 0
```

아래 그림은 PRACK 에 대한 ACK 입니다. PRACK 은 Request 이므로 이에 대한 처리 결과로써 200 OK 가 전송됩니다. 만일 여기서 UAS 가 200 OK 를 보내지 못하거나 전송간에 손실이 되었다면, 어떻게 될까요. PRACK 은 일반 Request 와 동일하므로 재전송이 발생합니다. ^^ 당연한 얘길 것입니다.

```
SIP/2.0 200 OK sip:bob@192.168.10.20
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asi98JK ;received=10.1.3.33
To: Bob <sip:bob@biloxi.com>; tag=a6c85e3
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 PRACK
Contact: <sip:alice@pc33.atlanta.com>
Content-Length: 0
```

7.3. Offer / Answer Model and PARCK

이미 제 3 장에서 다루었던 내용으로 early-session 과 session 에 대한 협상 시에 PRACK 을 통해 early-session 에 대한 응답을 수행했던 것을 상기 하시기 바랍니다. 이 부분에 대해 RFC 3262 에 나와 있는 좀 더 다양한 상황에 대해 정리하도록 하겠습니다.

일반적으로는 UAC 가 INVITE with Offer 를 제공하면, UAS 는 provisional response (예를 들면, 183 Session Progress) with Answer 로 응답하여 Final Response 이전에 세션 파라미터에 대한 협상이 진행됩니다. 만일 UAS 가 provisional response with Offer 를 한다고 가정한다면, 즉, UAC 가 INVITE without Offer 를 한 경우가 될 것입니다. 이때는 반드시 PRACK with Answer 가 되어야 합니다. 또한, PRACK with Offer 라면, PRACK 에 대한 200 OK 에 Answer 가 포함되어야 합니다. 이것은 기본적인 Offer / Answer Model 에 기초한 것으로 PRACK 을 통해 다양한 상황에서 세션 파라미터 협상이 가능함을 확인할 수 있는 부분입니다.

Chapter 8

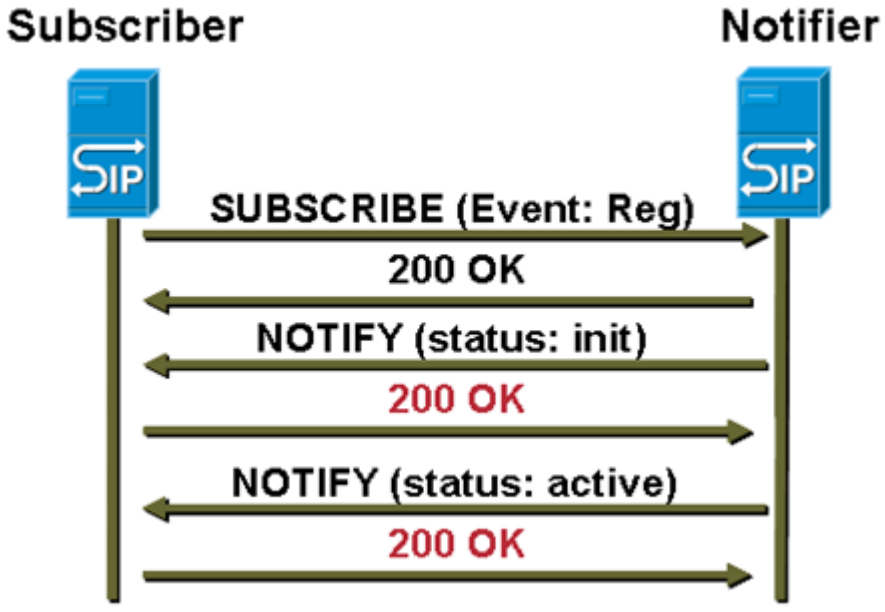
SUBSCRIBE 와 NOTIFY (RFC 3265,RFC 3680)

8.1. 개요

이벤트에 대한 통지 요청을 하고, 이에 대한 응답을 받는 다는 것은 다양한 SIP 응용 서비스에서 유용합니다. RFC 3265 SIP-Specific Event Notification 에 다음과 같은 SIP 응용 서비스를 예를 들고 있습니다.

- Automatic Callback Service (자동 콜백 서비스)
  특정 단말의 상태정보를 알 수 있도록 호가 종료되었을 때, 상태 정보가 SIP 서비스로 통지(Notification)된다면, 바로 콜백이 이루어질 것입니다. 콜백 서비스는 상대방이 부재중이거나 통화중일 때 송신자가 콜백 서비스를 신청해 놓으면, 상대방의 상태 변화가 감지되자마자 자동으로 통화가 되도록 하는 것입니다.
- Buddy Lists (친구 목록)
  버디 리스트에 등록된 친구 또는 동료의 상태정보가 실시간으로 통지되어 메신저와 같은 서비스를 편리하게 사용합니다.
- MWI (Message Waiting Indication)
  음성사서함의 상태 변화 이벤트가 실시간으로 전화 또는 메신저에 통지되어 음성 사서함의 저장된 음성을 들을 수 있습니다.
- PINT (PSTN and Internet Internetworking)
  PINT 에서도 호 상태 정보를 통한 서비스가 가능합니다. PINT 는 저도 처음 듣는 용어입니다. ^^ 인터넷에 찾아보니, "인터넷 응용을 요청하고 PSTN 전화서비스를 향상시키는 작업을 한다" 라고 합니다. 간단하게 유추해 보면, 인터넷 상의 응용 서비스가 PSTN 과 원활하게 호를 처리할 수 있도록 지능망과 연동하는 것이 아닌가 합니다.

이러한 SIP 응용 서비스는 SUBSCRIBE 와 NOTIFY 메쏘드를 활용합니다. 아래 그림은 호 또는 자원 (단말 등)의 상태를 Subscription 하고 상태의 변화를 통지하는 일반적인 절차입니다.



Subscriber 는 메신저 서버, UA 등 다양한 형태가 될 것이며, Notifier 는 Registrar 서버 또는 UA 입니다. 여기서 SUBSCRIBE 와 NOTIFY 가 언제 생성되는 지 알아 보겠습니다. SUBSCRIBE 메쏘드는 Subscriber 가 특정 사용자의 현재의 상태와 상태 정보 업데이트를 요청하여 향후의 이벤트 발생 시에 통지해 줄 것을 요청하는 것입니다. 또한, NOTIFY 는 SUBSCRIBE 의해 요청된 이벤트 발생시 통지하거나, Subscription (신청)된 단말의 상태 변화를 Subscriber 에게 통지합니다.

SUBSCRIBE 메시지 분석

위의 호 절차에 따라 패킷을 분석해 보도록 하겠습니다. 최초 Subscriber 가 Notifier 에게 SUBSCRIBE 요청을 합니다.

위의 SUBSCRIBE 요청에서 관심 있게 보아야 할 것은 다음과 같습니다.

- Event:reg
  SUBSCIBE 요청에 이벤트가 포함되어야 합니다. 이벤트는 엘리스의 등록 상태 정보를 요청하는 것입니다.
- Expires:21600
  SUBSCRIBE 요청에 이 헤더가 포함되어야 하며, 이 요청의 유효 기간을 명시합니다. 유효 기간 만료 전에 같은 다이얼로그(같은 Call-ID) 로 SUBSCRIBE 를 주기적으로 생성해야 합니다. 만일 Expires:0 로 설정한다면, Unsubscribe 를 요청하는 것입니다.

위의 SUBSCRIBE 는 정리하면, app_IM.atlanta.com (IM 서버)은 alice@atlata.com 의 등록 상태 및 21600 초 동안 발생되는 등록 상태 변화 이벤트 시에 통지해 줄것을 요청하는 것입니다.

위의 SUBSCRIBE 메시지가 SIP Registrar 서버에 요청이 된 후 아래와 같이 200 OK 메시지가 전송됩니다.

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP app_IM.atlanta.com
 ;branch=z9hG4bKnashds7 ;received=10.1.3.2
From: sip:app_IM.atlanta.com ;tag=123aa9
To: sip:alice@atlanta.com ;tag=xyzygg
Call-ID: 9987@app_IM.atlanta.com
CSeq: 9887 SUBSCRIBE
Contact: sip:server19.atlanta.com
Expires: 3600
```

200 OK 에서 Expires:3600 으로 보내졌습니다. 기존의 요청된 유효기간 보다 매우 짧은 3600 초 동안만 Registration 상태 정보에 대해 통지하겠다는 의미입니다. 이제 Notifier 에 의해 Accept 되었으므로 엘리스의 등록 상태 정보의 변화는 Subscriber 에게 한 시간 동안 통지될 것입니다. 엘리스가 등록되어 있음을 NOTIFY 매쏘드를 이용하여 보냅니다.

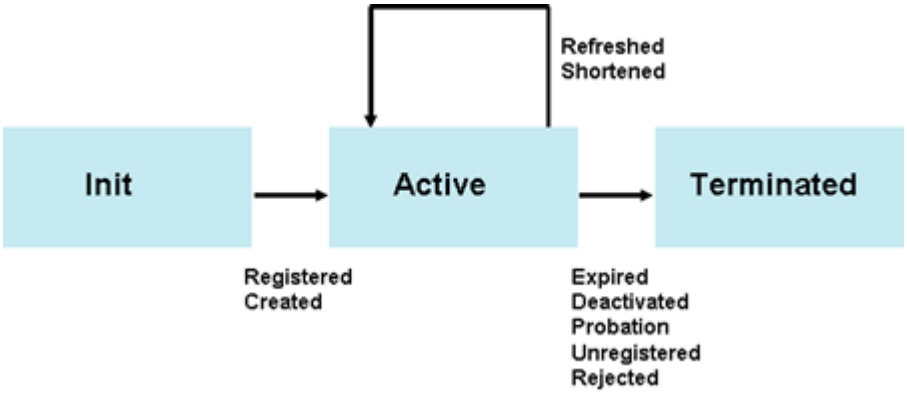
```
NOTIFY sip:app_IM.atlanta.com SIP/2.0
Via: SIP/2.0/TCP server1.atlanta.com
 ;branch=z9hG4bKnasaii
From: sip:alice@atlanta.com ;tag=xyzygg
To: sip:app_IM.atlanta.com ;tag=123aa9
Max-Forwards: 70
Call-ID: 9987@app_IM.atlanta.com
CSeq: 1288 NOTIFY
Contact: sip:server19.atlanta.com
Event: reg
Subscription-State: active
Content-Type: application/reginfo+xml
Content-Length: 223
<?xml version="1.0"?>
  <reginfo xmlns=
          "urn:ietf:params:xml:ns:reginfo"
           version="0" state="full">
    <registration aor="sip:alice@atlanta.com"
           id="a7" state="init" />
  </reginfo>
```

reginfo+xml 메시지에서 (노란색 부분) 엘리스의 등록 상태가 init 라고 표기하고 있습니다. 여기서 Subscription-State 헤더를 살펴보도록 하겠습니다. 이름에서 알 수 있듯이 Subscription 에 대한 상태 정보를 나타냅니다.

- Active
  Subscription 은 Accepted 및 Authorized 되었음을 의미
- Pending
  Notifier 가 Subscription 을 수령했지만, 불충분한 정책정보로 인해 승인 또는 거절 여부를 아직 결정하지 못함을 의미
- Terminated
  Subsription 이 종료되었음을 의미
  Expires 유효기간이 만료되었을 수도 있으며, 사유가 명기됨

8.2. Registration State Machine

RFC 3680 SIP Event Package for Registrations 에 따른 등록 상태의 변화에 대해 살펴보겠습니다. 아래 그림은 등록 상태 변화를 도식화한 것입니다.



이 과정을 이해하기 위해서 다시 한번 SIP Registration 을 짚어보겠습니다. SIP 에서 Registration 이란 Contact Address 와 Address-of-record 를 바인딩 하는 것입니다. 여기서 Contact Address 는 AoR 에 의한 식별된 사용자에게 연결되기 위한 실제 단말의 주소라고 할 수 있습니다. 이러한 일련의 과정을 위해 User Agent 가 SIP REGISTER 메쏘드를 이용합니다. 다음은 등록 상태 변화를 나타낸 것입니다.

- Init
  Address-of-record 에 등록된 contact 이 없는 상태 즉, 도메인에 등록된 사용자이나 통화하기 위한 Contact 주소가 확보되지 않았음을 의미합니다.
- Active
  하나 이상의 Contact address 가 바인딩 된 상태입니다.
- Terminated
  더 이상의 Contact address 가 존재하지 않는 상태로 바로 Init 상태로 전환됩니다.

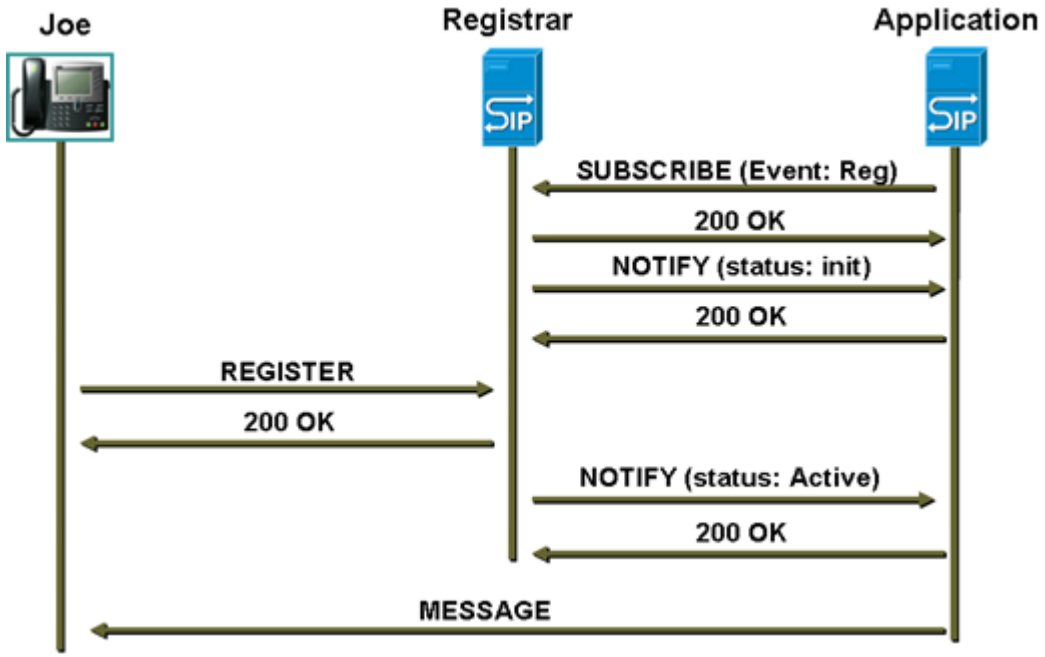
8.3. User Presence & Registration State

등록 상태 정보는 사용자 상태 정보 (Presence)와는 틀린 것입니다. 사용자 상태 정보는

사용자가 지금 망에서 다른 사용자와 통화할 수 있는지를 의미하며, 사용자와 연결하기 위한 다양한 통신 수단을 나타내는 Contact address 의 집합을 나타냅니다. 등록 상태 정보는 사용자와 연결하기 위한 Contact address 가 존재하는 지를 나타내는 것입니다. 즉, 사용자 상태 정보를 알기 위해서는 등록 상태 정보라는 Raw 데이터가 필요하게 되는 것입니다. 어렵게 설명된 듯합니다만, 찬찬히 읽어보시면 이해가 되실 것입니다.

8.4. "Welcome Notice "서비스 예제

RFC 3680 SIP Event Package for Registration 에는 등록 상태 정보를 이용한 몇 가지 시나리오가 있습니다. 이 중에서 RFC 3680 의 Welcome Notice 서비스 예제를 살펴보도록 하겠습니다. Welcome Notice 는 현재 일반적으로 사용되는 것으로 사용자가 로밍폰으로 외국에서 전화기를 켜면, 사용자의 단말이 Welcome to the country 메시지를 받게 됩니다. 이의 호 절차는 다음과 같습니다.



Application 은 특정 사용자 (Joe)의 등록 상태 정보를 요청하기 위해 Notifier 로 동작하는 Registrar 서버에 SUBSCRIBE 를 아래와 같이 전달합니다.

```
SUBSCRIBE sip:joe@example.com SIP/2.0
Via: SIP/2.0/UDP app.example.com;branch=z9hG4bKnashds7
From: sip:app.example.com;tag=123aa9
To: sip:joe@example.com
Call-ID: 9987@app.example.com
CSeq: 9887 SUBSCRIBE
Contact: sip:app.example.com
Event: reg
Max-Forwards: 70
Accept: application/reginfo+xml
```

Registrar 는 아래와 같이 200 OK 를 통해 3600 초 동안 Joe 의 등록 상태 정보를 보내겠다고 합니다.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP app.example.com;branch=z9hG4bKnashds7;received=192.0.2.1
From: sip:app.example.com;tag=123aa9
To: sip:joe@example.com;tag=xyzygg
Call-ID: 9987@app.example.com
CSeq: 9987 SUBSCRIBE
Contact: sip:server19.example.com
Expires: 3600
```

Registrar 는 200 OK 를 전송한 후 Joe 의 현재 등록 상태를 Application 에 NOTIFY 메쏘드를 통해 아래와 같이 전달합니다. reginfo+xml 포맷으로 된 Joe 의 등록 상태는 init 입니다. 즉, 사용자의 등록 정보는 있으되, Contact Address 가 아직 업데이트 되어 있지 않다는 것입니다.

```
NOTIFY sip:app.example.com SIP/2.0
Via: SIP/2.0/UDP server19.example.com;branch=z9hG4bKnasaii
From: sip:joe@example.com;tag=xyzygg
To: sip:app.example.com;tag=123aa9
Call-ID: 9987@app.example.com
CSeq: 1288 NOTIFY
Contact: sip:server19.example.com
Event: reg
Max-Forwards: 70
Content-Type: application/reginfo+xml
Content-Length: ...
<?xml version="1.0"?>
 <reginfo xmlns="urn:ietf:params:xml:ns:reginfo"
          version="0" state="full">
  <registration aor="sip:joe@example.com" id="a7" state="init" />
 </reginfo>
```

Joe 는 로밍 지역에서 전화기의 전원을 켜게 되면, 아래와 같이 등록 메시지가 전달됩니다.

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/UDP pc34.example.com;branch=z9hG4bKnaaff
From: sip:joe@example.com;tag=99a8s
To: sip:joe@example.com
Call-ID: 88askjda9@pc34.example.com
CSeq: 9976 REGISTER
Contact: sip:joe@pc34.example.com
```

Joe 가 Contact address 를 등록하게 되면, NOTIFY 메쏘드가 전달되게 되고, reginfo+XML 포맷을 보시면, 현재 Joe 가 Active 상태로 전환되었다고,

```
NOTIFY sip:app.example.com SIP/2.0
Via: SIP/2.0/UDP server19.example.com;branch=z9hG4bKnasaij
From: sip:joe@example.com;tag=xyzygg
To: sip:app.example.com;tag=123aa9
Call-ID: 9987@app.example.com
CSeq: 1289 NOTIFY
Contact: sip:server19.example.com
Event: reg
Max-Forwards: 70
Content-Type: application/reginfo+xml
Content-Length: ...
```

```
<?xml version="1.0"?>
 <reginfo xmlns="urn:ietf:params:xml:ns:reginfo"
        version="1" state="partial">
  <registration aor="sip:joe@example.com" id="a7" state="active">
   <contact id="76" state="active" event="registered"
        duration-registered="0">
     <uri>sip:joe@pc34.example.com</uri>
   </contact>
  </registration>
 </reginfo>
```

아래와 같이 Welcome to the example.com service! 메시지가 전송됩니다.

```
MESSAGE sip:joe@pc34.example.com SIP/2.0
  Via: SIP/2.0/UDP app.example.com;branch=z9hG4bKnashds8
  From: sip:app.example.com;tag=123aa10
  To: sip:joe@example.com
  Call-ID: 9988@app.example.com
  CSeq: 82779 MESSAGE
  Max-Forwards: 70
  Content-Type: text/plain
  Content-Length: ...

  Welcome to the example.com service!
```

Chapter 9

INFO (RFC 2976)

9.1. 개요

지금까지 세션을 설립하고, 필요한 세션 파라미터의 교환 등에 대해 중점적으로 다루었습니다. 세션이 설립된 후 기존의 세션을 유지하면서 필요한 정보를 교환하려면, 어떤 요청이 필요한지 생각해 봅시다. 기존의 메쏘드는 설립의 설립, 종료에 대한 것이었습니다. 200 OK 이후부터 BYE 이전까지 기존의 세션 내에서 UAC 와 UAS 간에 정보 교환을 할 수 있는 방법이 없습니다.

세션이 생성된 후, 즉 200 OK 이후의 세션 관련 제어 정보는 INFO 를 사용합니다. SIP INFO 는 SIP Signaling 경로를 이용하여 어플리케이션 레벨의 정보를 전송하는 것이 목적입니다. 다양한 정보를 교환할 수 있지만, 다음과 같은 정보의 변경은 불가능합니다.

- SIP 호의 상태 변경
- 초기 설정된 세션 파라미터의 변경

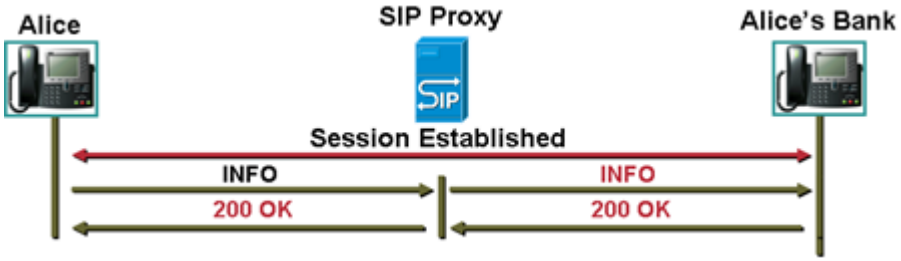
위와 같은 변경을 하기 위해서는 UPDATE 나 re-INVITE 를 통해 가능합니다. 단순히 세션 완료 전에는 UPDATE 를 세션 완료 후에는 re-INVITE 를 이용합니다. 물론, UPDATE 는 세션 완료 후에도 사용되지만, 추천하지 않습니다. 어쨌든 UPDATE 및 re-INVITE 는 다음 장에서 다루기로 하겠습니다. 여기서는 세션과 관련된 미디어 속성을 변경하거나 세션 타이머를 업데이트를 할 때는 UPDATE 나 re-INVITE 를, 어플리케이션 레벨의 세션 관련 제어 정보를 전송할 때는 INFO 를 사용한다고 이해하시면 되겠습니다.

SIP INFO 가 전송하는 주요 정보는 RFC 2976 The SIP INFO Method 에 다음과 같다고 명시합니다.

- PSTN 게이트웨이 간에 PSTN Signaling 메시지 전송
- DTMF Digits 전송
- WIreless Mobility 어플리케이션 지원을 위해 무선 신호의 세기를 전송
- Account balance 정보 전송 (선불카드 시스템에서 사용되는 것으로 추정)
- 세션 참가자간에 이미지 또는 비 스트리밍 정보를 전송

9.2. INFO 메쏘드의 호절차

항상 메쏘드에 대한 호 절차를 이해할 필요가 있습니다. SIP INFO 를 사용하는 절차는 다음과 같습니다.



세션이 설립 후 Alice 는 어플리케이션 레벨의 정보를 전송하기 위해 SIP INFO 를 전달합니다. 여기에서는 DTMF 를 전송한다고 가정합니다.

```
INFO sip:Alice's_Bank@192.168.10.20 SIP/2.0
Via: SIP/2.0/TCP pc33.atlanta.com
  ;branch=z9hG4bK776asegma
Max-Forwards: 70
To: Bank <sip:Bank@Bank_URI.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 22756 INFO
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: text/plain
Content-Length: 16

3 1 8 1 9 6 2
```

Alice's Bank 는 IVR 이라고 생각하면 됩니다. 세션 설립 후 IVR 이 주민 번호 또는 회원번호를 요구한 것이라고 생각할 수 있습니다. Alice's Bank 는 기존의 Call-ID 와 동일하므로, SIP INFO 를 받아들이고, 200 OK 를 전송합니다. RFC 2976 에는 Content-Type 에 대해 정의하지 않기에 필요에 따라 사용할 수 있습니다. 여기에서는 메시지 바디가 text/plain 으로 되어 있다고 나타나며, 전송된 정보는 Digits 3181962 입니다. 이에 대해 정상적으로 처리 되었으므로 200 OK 가 전송되었으며, 아래와 같이 다양한 응답이 발생할 수 있습니다.

- 481 Call leg / Transaction Dose not Exist
  만일 INFO 요청을 받은 UAS 가 기존의 Call leg 와 매치가 되지 않을 때
- 415 Unsupported Media Type
  UAS 가 이해할 수 없는 메시지 바디를 포함했으므로, 처리할 수 없을 때
- 200 OK (정상)
  UAS 가 이해할 수 있는 메시지 바디를 포함했고, 처리할 때
- 487 Request Terminated
  SIP INFO 요청을 처리 중인 가운데 CANCEL 메쏘드를 받았을 때

9.3. DTMF 전송의 개요

SIP INFO 를 말하면서 DTMF 전송을 말하지 않고 넘어갈 수 없습니다. 이 부분은 Sunnyside81@naver.com 님의 "VoIP 상에서의 DTMF 전송 비교" 라는 글을 참조하여 작성하였습니다. 언젠가 제가 다운로드를 받아놓은 자료입니다. 언제나 다시 읽어도 좋은 자료입니다. 인터넷에서 찾아서 한 번 읽어보시면, DTMF 에 대한 전체적인 이해에 도움이 되실 것입니다.

SIP 를 통한 DTMF 전송방식은 Out of band 와 In band 방식으로 구분됩니다.

- Out of band
  Signaling 메시지 및 Signaling Path 를 사용

DTMF duration 에 대한 정보가 없으므로 사용자가 Digit 을 길게 또는 짧게 누르는 것을 표현할 수 없음
SIP INFO 가 대표적

- In band
  RTP 메시지를 사용
  사용자가 Digit 을 길게 또는 짧게 누르는 것을 표현할 수 있음
  Bypass 및 RFC 2833 이 대표적

Out of band 를 사용하는 INFO 에 대해서는 위의 호 절차에서 설명하였습니다. Out of band 의 경우 DTMF Duration 을 전달할 수 없으므로, 사용자가 길게 Digit 버튼을 누를 때 발신자는 Tone 을 누른 기간만큼 듣지만, 수신자는 아무 소리도 들을 수 없습니다. 발신자가 Digit 버튼을 떼는 순간 Signaling 을 통해 메시지가 전달되어 수신자가 Tone 을 들을 수 있습니다.

In band 의 Bypass 방식은 Digit 을 RTP 가 사용하는 코덱으로 압축해서 음성과 같이 보내는 것입니다. 압축 코덱을 사용할 경우(g.711 이 아닌 g.723, g.729 등) DTMF tone 이 변형되거나 정보 손실이 발생할 가능성이 있습니다. 이러한 가능성이 상존하므로 잘 사용하지 않지만, 실제에서는 거의 정보 손실이 없었습니다. RFC 2833 을 이용한 방식은 RTP 패킷에 DTMF 의 번호와 볼륨, duration 이 명시되어 전송되므로 정보 손실의 가능성이 적지만, UDP 기반에서 동작하므로 패킷이 분실될 수 있기에 하나의 digit 을 여러 번 전송하여 정보 손실 가능성을 낮춥니다.
RFC 2976 에서 보듯이 INFO 의 Body 에 Digit 을 실어 보낼 수 있다고 되었지만, 정확한 형식에 대한 규정은 없어 제조사별로 고유한 방식을 사용합니다. 따라서, 서로 다른 제조사의 장비를 연결할 경우 SIP INFO 는 항상 문제를 일으킬 수 있습니다. 아래 두 개의 SIP INFO 메시지를 보시면 최초에 설명한 Content-Type 헤더 부분의 정의가 모두 다름을 알 수 있습니다.

```
INFO sip:0263121725@10.1.1.101:5060 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.101:5060;branch=z9hG4bK.dlsrks00000.46695
From: <sip:0263121725@hti.com>;tag=1103321111126510229
To: <sip:0263121726@hti.com:5060>;tag=18479239141126510229
Call-ID: 18073a010165400013341@10.1.1.101
CSeq: 23 INFO
Max-Forwards: 70
User-Agent: CallGenerator/1.0.0
Content-Type: audio/telephone-event
Content-Length:    4
INFO sip:0263121725@10.1.1.101:5060 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.101:5060;branch=z9hG4bK.dlsrks00000.313935
From: <sip:0263121725@hti.com>;tag=1103321111126510229
To: <sip:0263121726@hti.com:5060>;tag=18479239141126510229
Call-ID: 18073a010165400013341@10.1.1.101
CSeq: 22 INFO
Max-Forwards: 70
User-Agent: CallGenerator/1.0.0
Content-Type: application/vnd.networks.digits
Content-Length:    35
```

In-Band 로 DTMF 를 전송할 경우, Bypass 와 RFC 2833 두 가지 방법이 있다고 말씀 드렸습니다. Bypass 는 별도의 형식이나 정의가 필요 없습니다. 아무런 정의가 없으면, Bypass 로 전달합니다. RFC 2833 은 메시지 바디에 다음과 같은 내용이 포함됩니다.

```
v=0
o=0263121725 1126516338 1126516338 IN IP4 10.1.1.101
s=-
c=IN IP4 10.1.1.101
t=0 0
m=audio 30004 RTP/AVP 4 101
a=RTPmap:4 G723/8000
a=RTPmap:101 telephone-event/8000
```

위의 m 필드를 보시면, Payload Type 4 는 사용할 코덱에 대해서는 G723 을, Payload Type 101 은 Telephony-event 를 정의합니다. 이 Payload Type 101 을 이용하여 Digits 을 전달합니다. RFC 2833 은 같은 RTP 세션을 하지만, 서로 다른 Payload Type 을 사용하여 구분합니다. Payload Type 에 대해서는 "RTP 패킷 분석" 블로그를 참조하시기 바랍니다.

아래는 Bypass 를 사용할 경우의 메시지 바디입니다. 일반적인 코덱 정보 외에는 다른 협상정보가 없습니다. 따라서, 이럴 경우 Bypass 를 사용하는 것입니다.

```
v=0
o=0263121725 1126516338 1126516338 IN IP4 10.1.1.101
s=-
c=IN IP4 10.1.1.101
t=0 0
m=audio 30004 RTP/AVP 4
a=RTPmap:4 G723/8000
```

9.4. DTMF 전송 시의 에러

이러한 DTMF 전송은 IVR (Interactive Voice Response)과 같은 자동 응답 시스템에서 많이 사용됩니다. 일반적으로 IVR 은 수신된 Digits 을 재 확인하는 절차를 반드시 거치므로 전송간에 문제가 발생하더라도 사용자가 다시 Digits 을 송출하도록 하여 문제를 제거하는 절차가 필요합니다.

Out of band signal 을 통해 DTMF 를 전달할 경우 Inband 로는 DTMF tone 을 전송하지 말아야 하나 어떤 단말기는 사용자가 digit 버튼을 누르는 시점의 초기에 DTMF 의 Inband tone 을 보내기도 한다. 이것은 단말기가 DTMF 신호를 감지하면 해당 RTP 를 제거해야 하는데 감지하는 속도가 떨어지면 초기의 DTMF tone 이 RTP 를 통해 나간 후에 제거가 시작되는 데서 발생하는 것이다. 이러한 경우 수신한 Gateway 에는 Inband 로 들어온 DTMF tone 도 전송하고 Outband 로

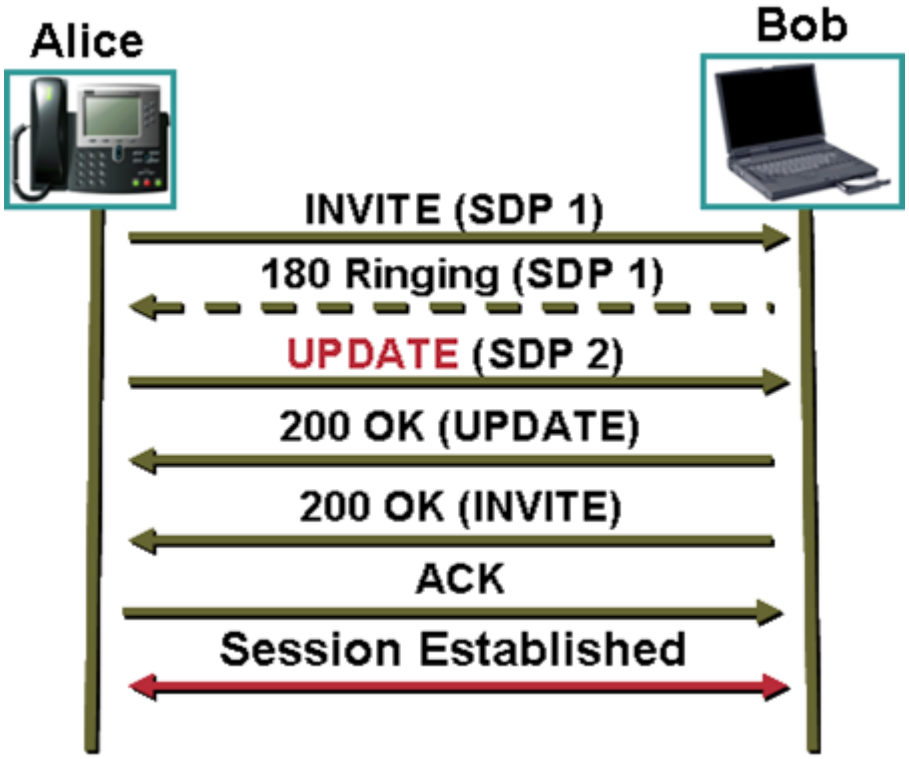
들어온 시그널도 DTMF tone 으로 변경하여 전송하므로 이 두 가지가 시차를 가지고 들어오는 경우 같은 digit 를 두 번 전송하는 결과를 초래합니다.

Chapter 10

UPDATE (RFC 3311)

10.1. 개요

이미 협상된 세션 파라미터를 변경하기 위한 re-INVITE 가 있는 데 왜 UPDATE 메쏘드가 정의되었는지 생각해 볼 필요가 있습니다. 아래 그림과 같이 엘리스가 INVITE 를 전송한 후 Ringback tone 을 듣고 있다가 밥이 수화기를 들기 전에 (200 OK 가 전송되기 전에) Hold 서비스를 호출했다고 가정해 봅시다. 즉, 기존의 다이얼로그가 그대로 유지되면서 Hold 서비스가 되어야 합니다. re-INVITE 를 통해 Hold 서비스를 호출할 수 있지만, 기존의 다이얼로그를 유지할 수 없으며, 별도의 세션이 설립되는 것입니다. 따라서, UPDATE 메쏘드를 이용하여 INVITE 메쏘드에 대한 Final Response (200 OK) 이전에 세션 파라미터를 업데이트하는 것입니다.



UPDATE 메쏘드는 기존에 협상된 미디어 스트림이나 코덱과 같은 세션 파라미터를 변경하기 위해 사용되지만, 기존의 다이얼로그를 그대로 유지 합니다. re-INVITE 는 마찬가지로 세션 파라미터를 변경하는 것은 같지만, 세션이 설립된 후에 사용되며 기존의 다이얼로그에 영향을 주게 되는 것이 차이점입니다.

여기서 Hold 서비스에 대해 잠깐 짚고 넘어 가겠습니다. 사용자가 전화기의 Hold 버튼을 누를 때, 단방향 a=sendonly 스트림으로 변경할 것을 요구하는 Offer 가 생성되고, 사용자의 전화기는 Mute (묵음) 상태가 되며, 상대측으로 미디어가 전송되지도, 로컬에서 미디어가 재생되지도 않습니다. 이 Offer 에 Media 에 대한 주소는 0.0.0.0 으로 설정합니다. Hold 는 RFC 3264 An Offer Answer Model with the SDP 에 자세히 나와 있습니다. 양방향 스트림을 단방향 스트림으로 전환하면서 상대방을 대기 상태로 만드는 것입니다. Hold 서비스는 Call Transfer 또는 Ad hoc Conference 를 하기 위해 많이 사용되는 사용자 기능입니다.

10.2. 호 절차

최초 INVITE 요청 시에 Allow 헤더에 UPDATE 를 명기하여 UAC 가 UPDATE 메쏘드를 지원함을 UAS 에게 알려줍니다. UAS 또한 180 Ringing 의 Allow 헤더에 UPDATE 를 명기하여 UPDATE 메쏘드를 지원함을 UAC 에게 알려줍니다. INVITE with OFFER (G.711)에 대해 180 Ringing with Answer (G.711)로 응답했습니다.

```
INVITE sip:bob@biloxi.com/TCP SIP/2.0
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928
Call-ID: a84b4c76e66710@pc33.atlanta.com
Allow: UPDATE
CSeq: 22756 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Requires: 100rel
Content-Type: application/sdp
Content-Length: 142

(Alice's SDP not shown)
(but calls for the codec G.711)
```

```
SIP/2.0 180 Ringing
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asdhds
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928
Call-ID: a84b4c76e66710@pc33.atlanta.com
Allow: UPDATE
CSeq: 22756 INVITE
RSeq: 813520
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

(Bob's SDP not shown)
(suitable response with the codec G.711)
```

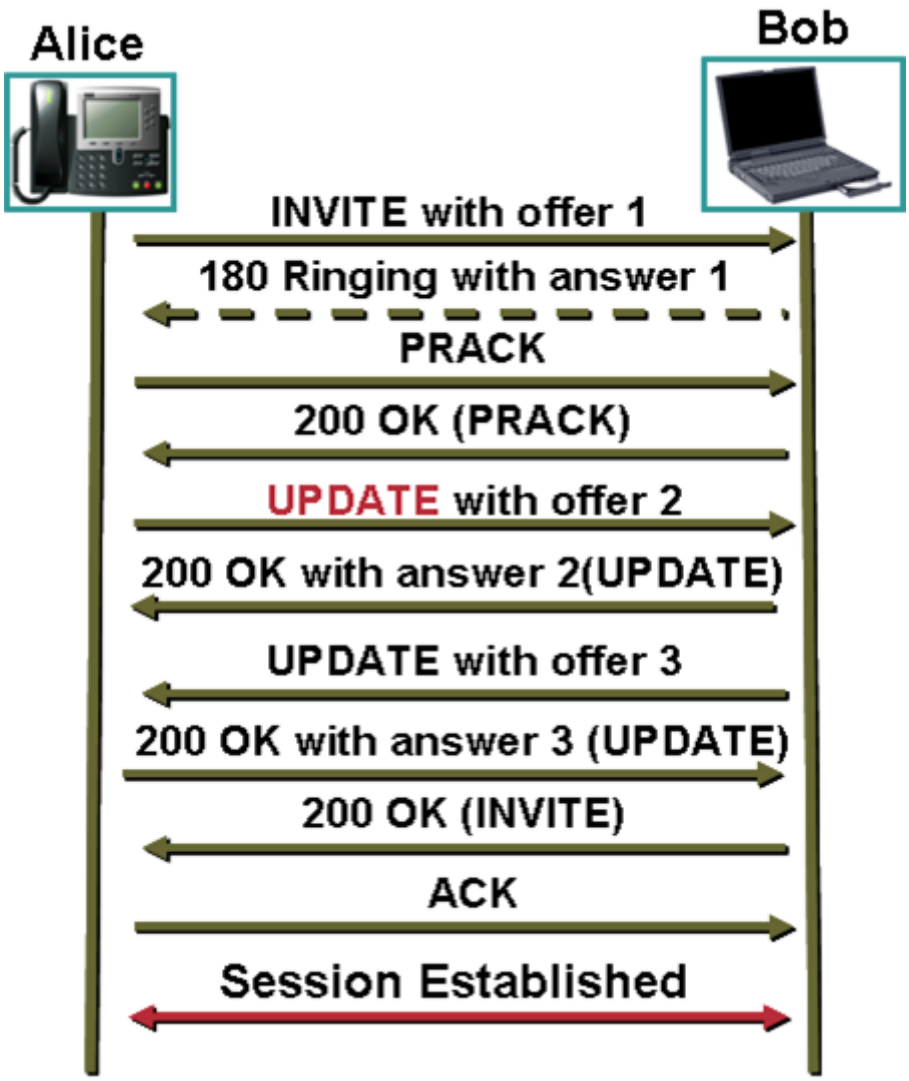
UPDATE 요청의 내용은 아래와 같습니다. 엘리스는 200 OK 전에 코덱을 G. 711 에서 G.729 로 변경할 것을 요청하며, 밥은 200 OK with answer (G.729)로 응답합니다.

```
UPDATE sip:bob@biloxi.com/TCP SIP/2.0
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 10197 UPDATE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142

(Alice's new SDP not shown)
(but calls for the codec G.729)
```

여기서 기억해야 할 것은 기존의 Offer / Answer 가 완료된 상태에서 UPDATE 메쏘드가 사용되어 세션 파라미터를 변경하는 것입니다. Offer / Answer 가 완료되지 않은 상태에서 UPDATE 메쏘드가 생성될 수 없습니다. INVITE without Offer 라면, 180 Ringing with Offer 가 전송되고, PRACK 을 통해 Answer 가 완료되어야 UPDATE 가 생성될 수 있습니다. 이 부분은 당연한 이야기인데 RFC 3311 The SIP UPDATE Method 에서 하도 길게 적어놔서 저도 한번 언급한 것입니다.

10.3. RFC 3311 Example Call Flow

백문이 불여일견 RFC 3311 에 나타난 호 절차를 보면서 다시 한 번 정리해 보도록 하겠습니다.



엘리스는 초기 INVITE 에 Offer 를 포함하여 전송하고, 밥은 180 Ringing 에 Answer 를 포함하여 응답을 합니다. 이 Provisional Response 에 신뢰성을 갖기 위해, 즉 Reliable Provisional Response 를 위해 PRACK 을 사용합니다. 이미 초기 INVITE 에 대해 프로세스가 완료되지 않았지만, 즉 200 OK 를 받지는 않았지만, 세션 파라미터 협상이 완료되었습니다. 이때, 엘리스는 Hold 서비스 (잠시 대기)를 호출하였고 이에 UPDATE 메소드가 전송됩니다. 두 번째 Offer 에서는 sendrecv 즉, 양방향 스트림을 UPDATE 매쏘드를 이용하여 단 방향 스트림으로 전환합니다. 즉, 밥이 스트림을 보내도 엘리스는 현재 듣지를 못하는 상태이기 때문입니다. 밥이 스트림을 양방향으로 다시 전환하기로 결정하고, UPDATE 를 통해 3 번째 Offer / Answer 를 수행합니다. 초기 INVITE 에 대한 Final Reponse 인 200 OK 를 전송하여 세션을 설립합니다.

Chapter 11

REFER (RFC 3515)

"CUCM 6.1 데이터 시트의 이해"를 약 1 년에 걸쳐서 연재를 하였지만, "SIP 의 이해"는 약 두 달 정도 만에 10 장까지 연재하였습니다. SIP 를 전체적으로 한 번 정리를 해야겠다는 생각에 빠르게 쓸 수 있었던 듯합니다. 이 연재가 저에게도 여러분들에게도 SIP 를 이해하는 데 도움이 되었으면 하는 바램입니다.

시작하기 전에

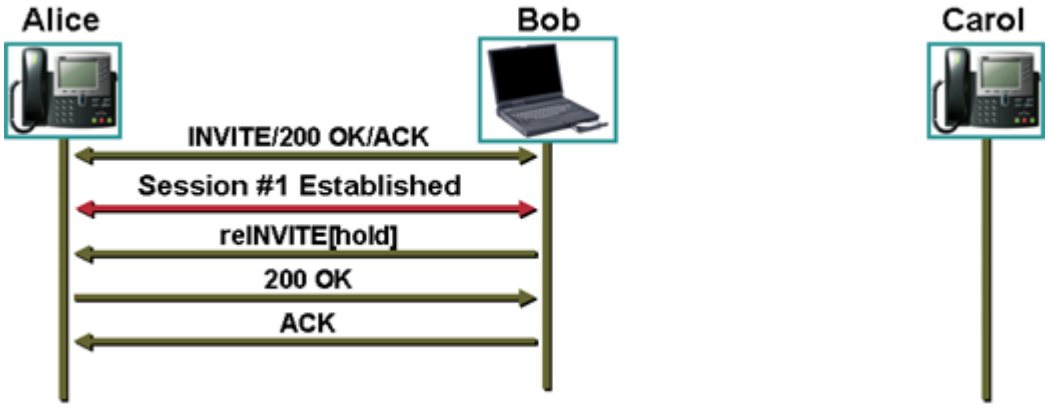
SIP 를 정리하면서 느끼는 점을 간단히 적어보면, SIP 는 이제 더 이상 간단한 프로토콜이 아니라고 생각됩니다. 다양한 서비스를 SIP 프로토콜 하나로 해결하려다 보니 매쏘드가 많아지고 있으며, 앞으로 더 많은 서비스가 SIP 로 구현될 것입니다. 지금 SIP 는 그 자체만으로도 매우 방대해졌으며, 단순히 VoIP 프로토콜로써 뿐만 아니라 UC 의 핵심 프로토콜로 성장하면서, SIP 도 여러 워킹 그룹에서 다루고 있습니다. 따라서, 더욱더 복잡해지고 있으며, 이는 이 기종 장비간 상호 연동성에 많은 장애가 됩니다. SIP 처럼 상호 연동에 어려움을 겪는 프로토콜이 없을 것입니다. 복잡성 외에도 SIP 의 유연성도 이 기종 장비간에 연동이 쉽지 않게 합니다. 유연성이라 한다면, 하나의 기능 구현에 있어서 이렇게 해도 되고, 저렇게 해도 되는 것이 아닌가 합니다. 대표적인 것이 Call Forward 일 것입니다. 302 Moved Temporarily 로 또는 181 Call Forwarded 로 응답을 하는 UA 에 대해 착신전환이 발생하는 것입니다. 어떤 장비는 302 Response 를 무시하여 호를 종료하기도 합니다.

제 생각에는 H.323 처럼 좀 더 엄격하게 표준이 정의 되어야 하는 데, 유연성의 장점을 버리지 못해 - 쉽게 버릴 수도 없는 - 이런 문제들이 발생합니다. 그러나, SIP 가 향후 VoIP 프로토콜을 주도할 것이므로 시간이 지나면 이런 문제들이 하나 둘씩 극복되지 않을 까 합니다.

11.1. 개요

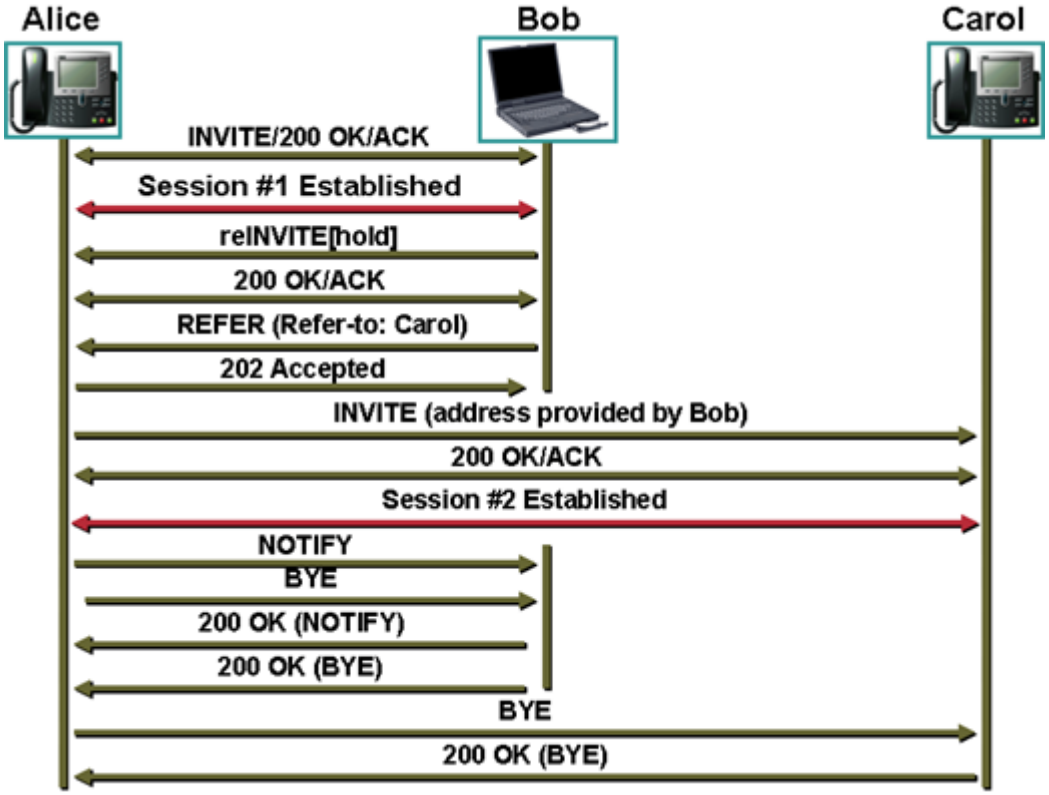
REFER 는 SIP 메쏘드로써, REFER 를 받는 수신 측은 Request 에 제시되는 리소스를 참조하도록 합니다. 많은 어플리케이션에서 사용될 수 있으며, Call Transfer 와 같은 서비스가 대표적인 사용 예입니다. 쉽게, 하나의 UA 가 다른 UA 에게 직접 INVITE 를 요청하도록 하는 것입니다. REFER 메쏘드를 받는 수신 측은 INVITE /200 OK / ACK 를 제 삼자와 통신한 후 NOTIFY 메쏘드를 통해 REFER 전송자에게 보고하도록 되어 있습니다. 또한, REFER 에 대한 응답 202 Accepted 를 사용합니다.

11.2. 호 절차 (Call Transfer)



위의 그림은 보시겠습니다. 엘리와 밥이 항상 통화를 했었는데 캐롤이 새로 등장합니다. 밥에게 새로운 애인이 생긴 모양입니다. ^^ 이제는 기본적인 INVITE 요청에서 ACK 까지는 설명 드리지 않아도 엘리스와 밥간에 호가 성립되어 통화중임을 알 수 있을 것입니다. 통화중에 밥은 엘리스를 캐롤과 통화하도록 연결해 주고자 합니다. 밥은 엘리스에게 "잠시만, 캐롤과 통화하도록 해줄게"라고 말하고 Hold 버튼을 선택할 것입니다.

Hold (통화 중 대기) 는 엘리스가 잠시 대기하도록 하는 것입니다. 이미 앞에서도 Hold 에 대해 말씀을 드렸었는 데 간단하게 정리하겠습니다. reINVITE with SDP 가 엘리스에게 전달되고, SDP 에는 a=sendonly , c=0.0.0.0 라고 표시가 됩니다. 200 OK with SDP 가 밥에게 전달되고, SDP 에는 a=recvonly 라고 표시가 됩니다. 만일 MoH (Music on Hold)가 사용된다면, c=0.0.0.0 이 특정 서버의 IP address 를 가리킬 것입니다.



위의 그림을 보면, Hold 를 한 후 Carol 의 정보를 REFER 를 통해 보내게 됩니다. REFER 을 수신한 엘리스는 INVITE 를 직접 Carol 에 요청합니다. 엘리스와 캐롤간에 통화로가 설정되고 나서 엘리스는 밥에게 현재의 상황을 NOTIFY 매쏘드를 통해 Bob 에게 보고합니다. 동시에 BYE 매쏘드를 밥에게 전송하여 엘리스와 밥간의 호는 종료되고, 엘리스와 캐롤간에 통화 후 종료하게 됩니다. 위의 상황을 본다면, Blind Transfer 가 되겠습니다.

밥이 엘리스에게 보낸 REFER 메시지는 아래와 같습니다.

```
REFER sip:alice@atlanta.com SIP/2.0
Via: SIP/2.0/UDP swp34.biloxi.com
  ;branch=z9hG4bKna9
Max-Forwards: 70
To: <sip:alice@atlanta.com>;tag=a6c85cf
From: <sip:bob@biloxi.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 10187 REFER
Allow: INVITE, ACK, CANCEL, OPTIONS,
  BYE, REFER, NOTIFY, UPDATE
Supported: replaces
Refer-To: <sip:carol@chicago.com>
Contact: <sip:bob@swp34.biloxi.com>
Content-Length: 0
```

여기서 Refer-To 헤더를 짚고 넘어가겠습니다. REFER 메시지에 반드시 포함되어야 하며, 엘리스가 INVITE 를 생성한 후 보낼 정확한 목적지가 명기됩니다. REFER 의 다양한 응용을 위해서는 Refer-To 헤더에 다양한 양식의 URI 입력이 가능해야 할 것입니다. 아래는 RFC 3515 에 나타난 예제입니다.

Refer-To: sip:alice@atlanta.example.com
Refer-To: <sip:bob@biloxi.example.net?Accept-Contact=sip:bobsdesk.
biloxi.example.net&Call-ID%3D55432%40alicepc.atlanta.example.com>
Refer-To: <sip:dave@denver.example.org?Replaces=12345%40192.168.118.3%3B
to-tag%3D12345%3Bfrom-tag%3D5FFE-3994>
Refer-To: <sip:carol@cleveland.example.org;method=SUBSCRIBE>
Refer-To: http://www.ietf.org

또한, NOTIFY 메시지는 아래와 같습니다. Event: refer 를 통해 이 NOTIFY 는 REFER 에 이벤트에 의해 생성됨을 확인할 수 있습니다. (아래 NOTIFY 메시지는 RFC 3515 를 참조하여 제가 직접 호 절차에 맞게 대충 내용을 맞추다 보니 틀릴 수 있는 부분이 있습니다. 이런 흐름이다라는 것으로 이해해 주시면 될듯합니다.)

```
NOTIFY sip:bob@atlanta.example.com SIP/2.0
Via: SIP/2.0/UDP agentb.atlanta.example.com;branch=z9hG4bK9922ef992-25
To: <sip:bob@biloxi.com>;tag=1928301774
From: <sip:alice@atlanta.com>;tag=a6c85cf
Call-ID: a84b4c76e66710@pc33.atlanta.comCSeq: 1993402 NOTIFY
Max-Forwards: 70
Event: refer
Subscription-State: active;expires = (depends on Refer-To URI)
Contact: sip:alice@atlanta.example.com
Content-Type: message/sipfrag;version=2.0
Content-Length: 20

SIP/2.0 200 OK
```

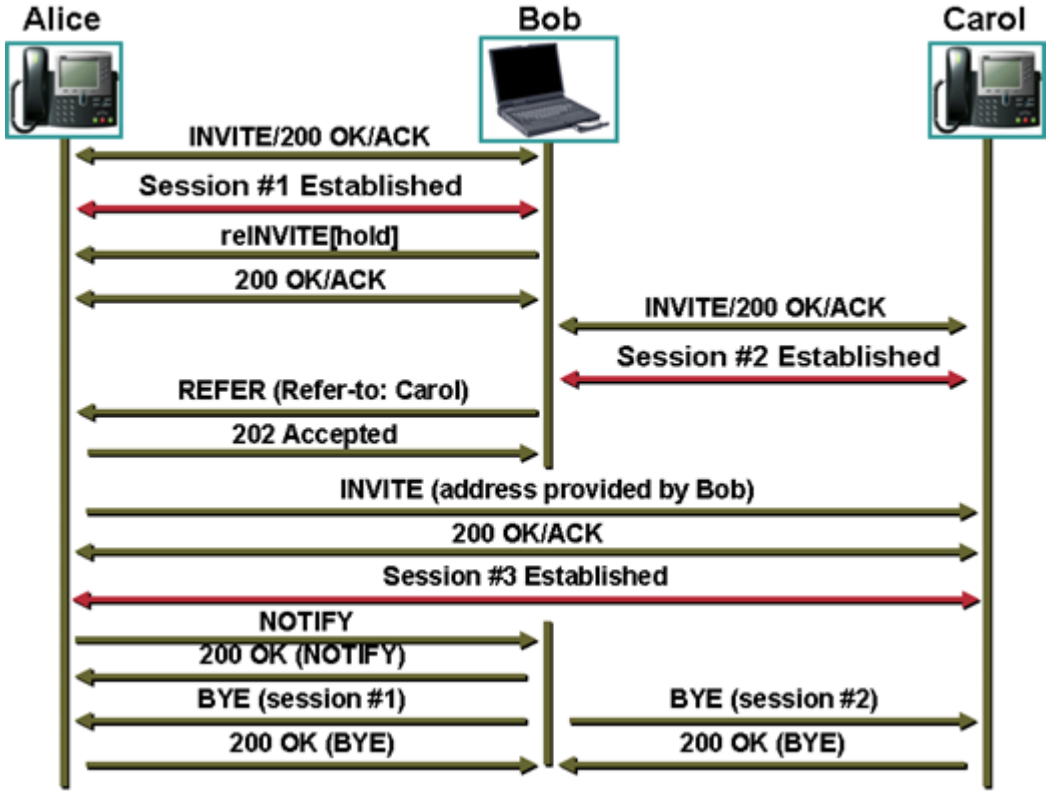
To, From, Call ID 가 정확하게 REFER 와 매치가 되어야 되는 것은 당연한 것일 것입니다. 또한 메시지 바디에 message/sipfrag 가 명시되어야 하며, 이는 REFER 에 의한 이벤트의 상태를 나타냅니다. 여기에서는 SIP/2.0 100 Trying 이라고 되어있습니다. 이는 다음과 같은 의미입니다.

- SIP/2.0 100 Trying
  현재 REFER 에 의해 요청된 이벤트가 처리중임
- SIP/2.0 200 OK
  현재 REFER 에 의해 요청된 이벤트가 성공적으로 처리되었음
- SIP/2.0 503 Service Unavailable
  현재 REFER 에 의해 요청된 이벤트가 실패했음

- SIP/2.0 603 Declined
  현재 REFER 에 의해 요청된 이벤트를 수신했지만, 거절되었음

Subscription-State 헤더를 통해 Carol 이 Active 임을 통지합니다. 이 헤더를 통해 나타냅니다. 그리고, 마지막 NOTIFY 가 전송될 때는 Subscription-State:terminatd;reason=noresource 가 반드시 명기되어야 합니다.

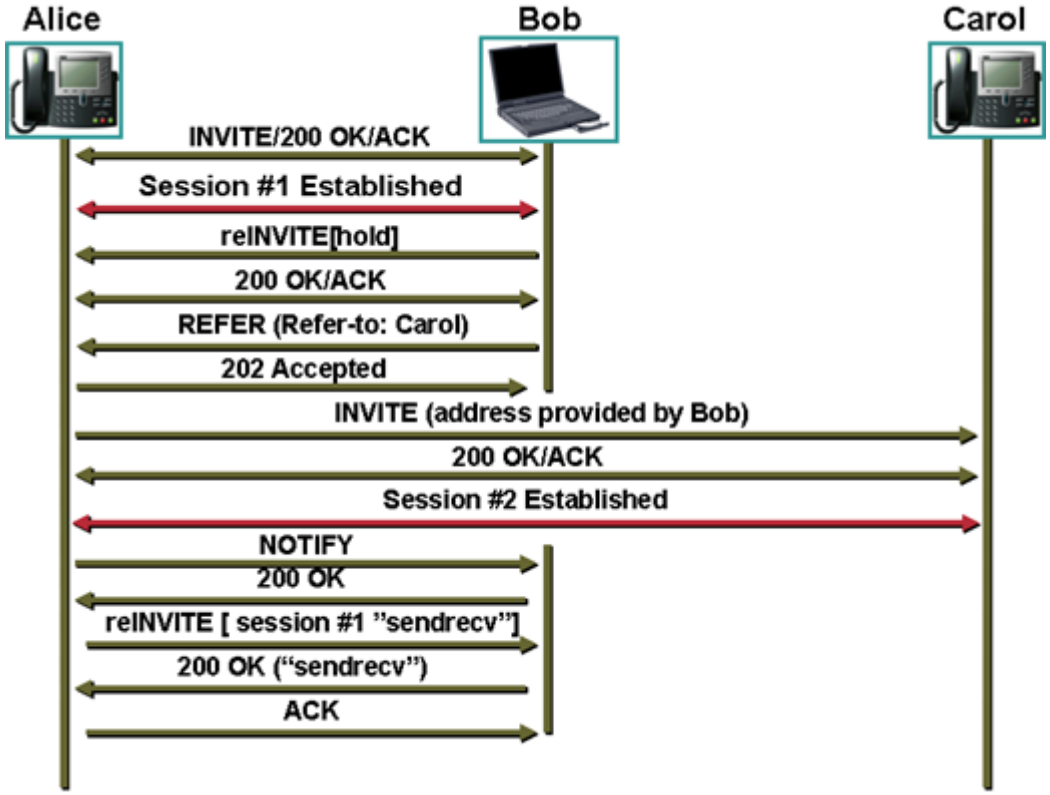
자 그럼 Call Transfer with consultative 는 어떻게 될지 생각해 보겠습니다. 이는 밥이 캐롤과 통화 후 캐롤에게 "잠시만 엘리스 전화 돌려줄께" 하고 Transfer 버튼을 누르게 될 것입니다. 이의 절차는 아래과 같습니다. 저는 항상 답을 금방 주는 편입니다. ^^



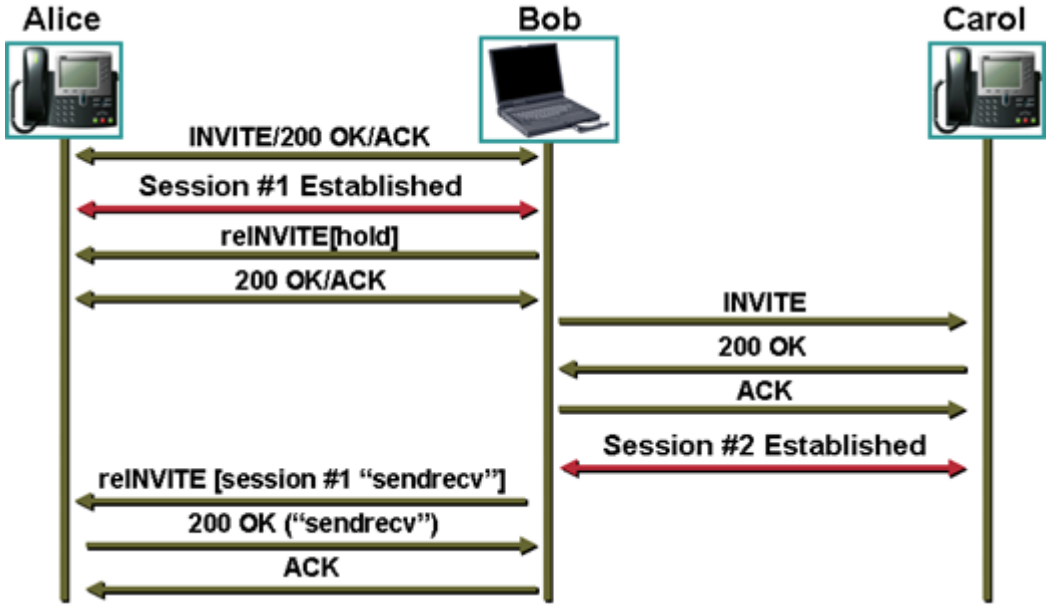
엘리스와 밥이 통화 후 밥은 엘리스를 Hold 시킨 후, 밥이 캐롤과 통화합니다. 그리고, 밥은 앨리스에게 REFER 를 전송하여 엘리스가 캐롤에게 직접 INVITE 를 전송하도록 합니다. 엘리스는 캐롤과 세션이 연결되자마자 NOTIFY 로 밥에게 보고합니다. 이 상태에서 밥은 엘리스와의 통화와 캐롤과의 통화 모두를 종료하게 되는 것입니다.

11.3. 호 절차 (3-way Conference)

REFER 를 이용한 또 다른 응용서비스로 아래 그림과 같이 삼자 통화를 들 수 있겠습니다.



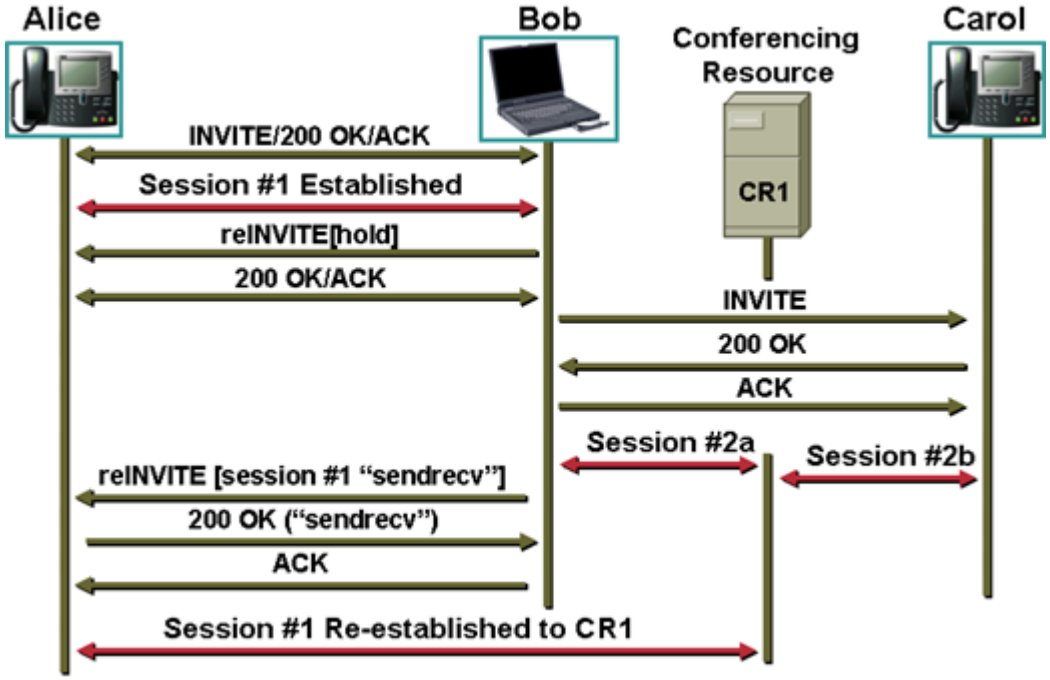
Call Transfer 와 3 자 통화의 차이라면, 엘리스가 캐롤과 호를 설립하고 나서, NOTIFY 를 밥에게 전송합니다. 밥은 엘리스에게 reINVITE 를 전송하여 기존의 Hold 상태의 세션을 양방향 통신 상태로 전환합니다. 즉, Call Transfer 에서는 BYE 로 기존 세션을 종료한 것과 달리 엘리스가 호를 유지하게 되고, 엘리스는 밥과 캐롤의 목소리를 믹싱하여 밥과 엘리스에게 전송하여야 합니다. 따라서, DSP 가 바쁘게 움직일 것입니다. 이 것은 엘리스에 의해 삼자 통화가 되는 것입니다. 일반적인 삼자 통화는 아래 그림과 같을 것입니다.



밥이 엘리스와 통화 중에 엘리스를 통화중 대기 시킨 후 다시 캐롤과 통화하게 되고 두 개의 호를 밥이 Join 하여 Ad-hoc Conference 를 하게 되는 것입니다. 삼자 통화는 Refer 를 이용할 수도 이용하지 않을 수도 있다는 것을 이해하시면 될 듯합니다.

11.4. 호절차 (3-way conference with MCU)

삼자 통화를 살펴보는 참에 MCU 또는 DSPFram 을 이용한 Conference 를 살펴보겠습니다. 만일, 통화자가 서로 다른 코덱을 사용하게 되면, 어쩔 수 없이 외부 리소스를 사용해야 합니다. 이를 사용하는 경우에 대한 호 절차를 살펴보겠습니다.



위의 호 절차를 보시면, 외부 리소스를 사용하여, 삼자 통화를 하는 것을 나타냅니다. 아마도 호 절차는 이해가 되는 데 RTP 가 어떻게 Conference Resource 로 묶이게 되는 지를 이해하지 못하시겠다면, 지금까지의 SIP 연재를 상기해 보시기 바랍니다.

11.5. 결론

처음 이 메쏘드를 정리할 때 RFC 3515 The SIP REFER Method 의 예제를 가지고 정리하려고 했는데 예제가 구체적인 상황을 명시하지 않아 이해하기가 난해하리라 생각되어 다른 예제를 찾아보다 삼자 통화 호 절차까지 다루게 되었습니다.

Chapter 12

PUBLISH (RFC 3903)

이번 글에서는 PUBLISH 메쏘드에 대해 살펴보겠습니다. 이 메쏘드는 8 장 SUBSCRIBE & NOTIFY 를 다룰 때 함께 다루었어야 하는 데 차례를 잘 못 정의한 관계로 뒤에서 다루게 되었습니다. 따라서, 잘 이해가 되지 않는 부분이 있으면, 8 장을 다시 한 번 읽어보시는 것이 글을 이해하는 데 도움이 되실 것입니다.
이 글은 RFC 3903 SIP Extension for Event State Publication 및 RFC 3856 A Presence Event Package for SIP 글을 참조하였습니다. 처음에는 RFC 3903 만 가지고 정리하다 보니 어딘지 모르게 2% 부족한 글이 되었습니다. 2%를 찾아 헤매다 보니 RFC 3856 을 발견하게 되었습니다. 두 개의 내용을 함께 정리하는 것이 좀 더 편하게 정리할 수 있다고 생각됩니다. 또한, RFC 3856 은 PUBLISH 메쏘드의 존재를 모르는 상황에서 presence 의 정보를 업데이트하는 구조로 되어 있기에 RFC 3856 의 부족한 2%를 채울 수 있는 것 같습니다.

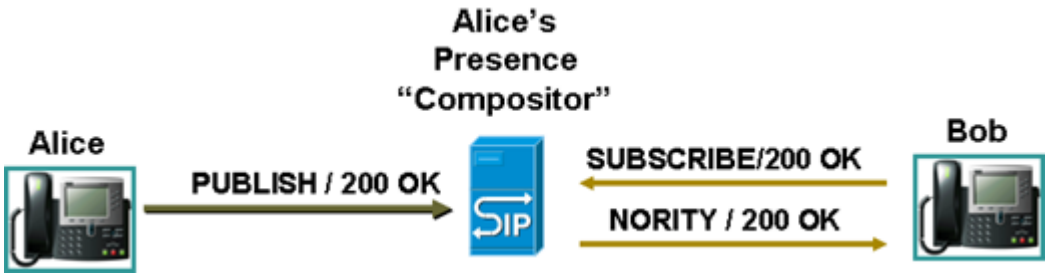
12.1. 개요

구글, 네이트온, MSN 등과 같은 메신저에 우리는 쉽게 상대방의 Presence (현재의 상태) 정보를 쉽게 확인할 수 있습니다. Presence 는 네트워크 상에서 통신을 위한 사용자의 상태정보 (willingness and ability)라고 정의할 수 있습니다. 이는 단순히 on-line 및 off-line 정보일 수 있으며, 우리가 쓰는 것처럼 회의 중 (Meeting), 통화 중 (Busy), 자리 비움 (Away)등과 같이 다양한 정보로 표현될 수 있습니다. 이러한 Presence 정보를 서로 주고 받기 위해 REGISTER, PUBLISH, SUBSCRIBE, NOTIFY 가 유기적으로 동작합니다.
RFC 2778 A Model for Presence and Instant Messaging 자료를 참조하시면, 기본적인 용어 및 메커니즘을 이해할 수 있습니다. 향후에 SIMPLE (SIP for Instant Messaging and Presence Leveraging Extension)을 꼭 다룰 것이 때문에 여기에서는 생략하도록 하겠습니다. ^^
RFC 3903 SIP Extension for Event State Publication 의 용어 정의를 통해 주요 구성 요소에 대해 살펴보겠습니다.

- Event State
  자원의 상태 정보
- EPA (Event Publication Agent)
  PUBLISH 요청를 발행하는 UAC , 아래 그림에서 Alice 의 역할
  RFC 3856 의 PUA (Presence User Agent)
- ESC (Event State Compositor)
  PUBLISH 요청을 수신 받아 처리하는 UAS, 아래그림에서 Compositor 의 역할
  RFC 3856 의 PA (Presence Agent) 로써, Proxy 및 Register 와 공존할 수 있음
- Event Hard State
  자원의 default Event State 로 AoR 에 대한 고정된 상태 정보
  ESC 는 Soft State publication 이 없을 때 사용

- Event Soft State
  PUBLISH 메커니즘을 통해 EPA 가 발행하는 Event State, 유효기간을 가지고 있으며, 만료 시에 재협상되는 유효기간 내에서만 의미 있는 Event State 를 나타냄

정리하면, PUBLISH 메쏘드는 AoR (Address-of-Record)과 연관된 Event State 를 생성, 변경, 제거하기 위한 메쏘드입니다. REGISTER 와 비슷하지만, REGISTER 와 달리 다이얼로그를 생성하지 않습니다. 따라서, 상태의 변화에 즉각적인 퍼블리싱이 가능하며, 기존의 다이얼로그에 영향을 미치지도 않습니다. 또한, REGISTER 를 통해 표시할 수 있는 Presence 정보는 제한적일 수 있지만, PUBLISH 메쏘드를 통해 좀 더 다양한 형태의 Presence 정보 교환이 가능합니다.



위의 그림에서 보듯이 Presence 를 수집하고, PUBLISH 요청을 처리하는 UAS 를 Compositor 즉 ESC 라고 합니다. Bob 은 Compositor 에 SUBSCRIBE 메쏘드를 통해 Alice 의 Presence 가 변경되면 알려줄 것을 요청하고, Alice 가 Logon 하여 Presence 정보를 Compositor 에 업데이트하면, Bob 에게 Alice 의 Presence 정보가 NOTIFY 를 통해 통지 됩니다.

12.2. 호 절차

이미 SUBSCRIBE 및 NOTIFY 의 동작에 대해서는 설명했으므로 생략하고, 아래 그림에서처럼 PUBLISH 메쏘드에 대해 살펴보겠습니다.

Alice 는 ESC 에 자신의 Presence 정보를 Publication 할 것이며, 이 때의 SIP 헤더는 아래와 같습니다. 생략된 XML 메시지 바디는 8 장의 NOTIFY 에 전송되는 내용과 동일한 형태를 취합니다. presence 에 대한 이벤트 패키지를 사용하므로 Event:presence 로 전달됩니다. REGISTER 메쏘드의 경우 Event:reg 로 되어 있던 것을 기억하시기 바라며, 두 메쏘드 모두 Expires 헤더를 가집니다. 아래 그림을 보시면, 상태정보를 나타내는 헤더가 없습니다. 이 내용은 NOTIFY 에서와 마찬가지로 XML 메시지 바디에 포함되어 있습니다.

```
PUBLISH sip:esc1@biloxi.com SIP/2.0
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asegma
Max-Forwards: 70
To: Alice <sip:alice@atlanta.com >
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 22756 PUBLISH
Event: presence
Expires: 21600
Content-Type: application/pidf+xml
Content-Length: 126

(XML message body not shown)
```

아래 그림은 PUBLISH 에 대한 200 OK 응답 메시지입니다. 새롭게 SIP-ETag 가 추가된 것을 확인할 수 있습니다.

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP pc33.atlanta.com
 ;branch=z9hG4bK776asegma ;received=10.1.3.33
To: sip: sip:bob@biloxi.com>;tag=1928301774
From: alice@atlanta.com
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 22756 PUBLISH
SIP-ETag: hp169abc
Expires: 1800
```

12.3. 마치며..

향후에 이 연재를 마치면, Presence 및 SIMPLE 에 대해 좀 더 깊게 다루도록 하겠습니다. 여기에서는 매쏘드 위주로 살펴보는 것으로 만족해야겠습니다. 연재를 하면서 나중에 다시 다루겠다는 표현을 몇 번 한 것으로 기억하는 데 다 지킬 수 있을 지가 의문입니다. ^^ 제가 하지 않더라도 다른 블로거들이 해주리라 믿으면서, 다음에는 연재의 마지막인 MESSAGE 에 대해 아주 간략하게 살펴보겠습니다.