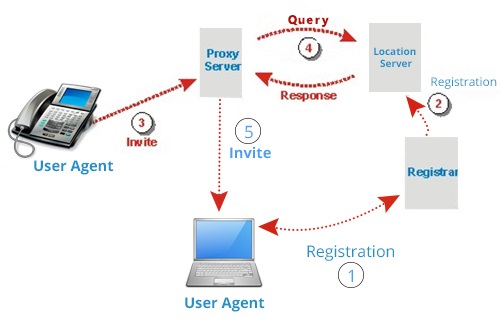
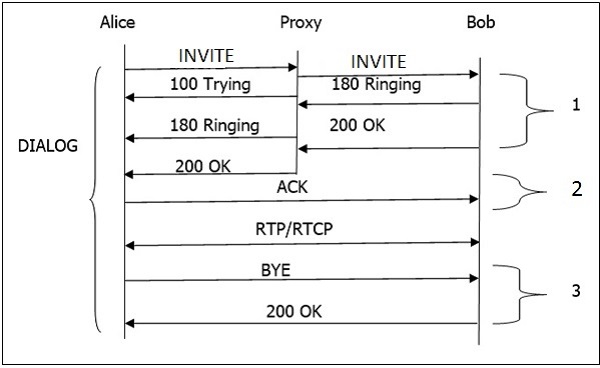
1. sip协议是基于文本的，所谓基于文本是指以\n区分内容。相比二进制协议的区别，二进制协议通过规定头的长度来区分头部和数据部
2. sip属于应用层协议，使用UDP以及TCP进行传输
3. SIP 的一个重要特点是它不定义要建立的会话的类型，而只定义应该如何管理会话。
4. sip组成：
5. **用户代理**：可以理解为终端。如电话等。可以创建、修改、终止会话，分为：
6. 用户代理客户端：发送请求并接收响应的实体（相当于呼叫者）
7. 用户代理服务器：接收请求并发送响应的实体(相当于被叫方)
8. **代理服务器**：相当于路由器，接收来自用户代理的请求并将其转发给另一个用户，位于两个用户代理之间，最多可以有70个
9. **无状态代理服务器** - 它只是转发收到的消息。这种类型的服务器不存储任何呼叫或交易的信息。
10. **有状态代理服务器** - 这种类型的代理服务器可以跟踪收到的每个请求和响应，并且如果需要，可以将来使用它。如果对方没有响应，它可以重新发送请求。
11. **注册服务器**：注册服务器接受用户代理的注册请求。它可以帮助用户在网络中进行身份验证。它将URI和用户的位置存储在数据库中，以帮助同一域内的其他SIP服务器。
12. **重定向服务器:** 重定向服务器接收请求，并在注册器创建的位置数据库中查找请求的预期收件人。
13. **位置服务器：**位置服务器提供有关呼叫者可能的位置到重定向和代理服务器的信息



1. sip的层级划分

sip协议被设计为分层协议。共划分为4层。由底层向上依次为：语法和编码层、传输层、事务层、事务用户层

1. SIP的最低层是其语法和编码。其编码使用增强的Backus-Naur表格语法（BNF）来指定。
2. 第二层是传输层。它定义客户端如何发送请求并接收响应，以及服务器如何接收请求并通过网络发送响应。所有SIP元素都包含传输层。
3. 接下来是事务层。事务是由客户端事务（使用传输层）发送到服务器事务的请求，以及从服务器事务发送回客户端的对该请求的所有响应。用户代理客户端（UAC）完成的任何任务都将使用一系列事务进行。无状态代理不包含事务层。
4. 事务层上面的层称为事务用户。除了无状态代理之外，每个SIP实体都是一个事务用户。
5. sip会话流程：



以下是对上述呼叫流程的逐步说明 -

* 发送到代理服务器的INVITE请求负责启动会话。
* 代理服务器发送**100 尝试**立即响应呼叫者（Alice）以停止INVITE请求的重新发送。
* 代理服务器在位置服务器中搜索Bob的地址。获取地址后，进一步转发INVITE请求。
* 此后，Bob手机生成的**180 振铃**（临时响应）返回给爱丽丝。
* 鲍勃拿起手机后一个**200 OK**响应很快产生。
* 一旦**200 OK**到达Alice，Bob 从Alice 收到一个**ACK。**
* 同时，会话建立，RTP数据包（会话）从两端开始流动。
* 会话结束后，任何参与者（Alice或Bob）都可以发送一个**BYE**请求来终止会话。
* **BYE**直接从Alice到Bob绕过代理服务器。
* 最后，Bob发送**200 OK**响应来确认BYE，会话终止。
* 在上述基本呼叫流程中，可以使用三个**事务**（标记为1，2，3）。

完整的呼叫（从INVITE到200 OK）称为**对话Dialog**。

1. SIP消息有两种类型 - **请求**和**响应**。
2. 请求的开始行包含定义请求的方法，以及定义要发送请求的请求URI。
3. 类似地，响应的开始行包含响应代码。

**SIP请求**是用于建立通信的代码。为了补充它们，**SIP响应**通常指示请求是成功还是失败。

SIP请求中包含请求的方法（类似消息类型？）包括核心方法和扩展方法。核心方法共有6个：

INVITE：INVITE用于启动与用户代理的会话

BYE：用于终止既定会话的方法

REGISTER：请求执行用户代理的注册

CANCEL：终止未建立的会话

ACK：用于确认对INVITE方法的最终响应

OPTIONS: 用于向用户代理或代理服务器询问其功能并发现其当前的可用性

扩展方法：

SUBSCRIBE：建​​立订阅，获取有关特定事件的通知

NOTIFY：获取特定事件的发生

PUBLISH：被用户代理用于向服务器发送事件状态信息

REFER：由用户代理用于引用另一个用户代理来访问对话框的URI

INFO由用户代理使用，以向其已经建立媒体会话的另一用户代理发送呼叫信令信息。

UPDATE:如果会话未建立，则UPDATE用于修改会话的状态.如果会话建立，则使用重新邀请来更改/更新会话。

PRACK:用于确认接收到可靠的临时响应转移（1XX）。

MESSAGE: 用于使用SIP发送即时消息。IM通常由参与文字会话的参与者实时交换的短消息

1. SIP协议格式

SIP协议不同的区域通过CRLF进行划分（基于文本的方式）

SIP消息分为请求和响应

消息结构：

generic-message = start-line

\*message-header

CRLF

[ message-body ]

start-line = Request-Line（对应请求） / Status-Line（对应应答）

1. **Request-Line: Method SP Request-URI SP SIP-Version CRLF**

* **Method**既上面所说的六种请求方法。
* **SP**表示空格
* **Request-URI**: 即SIP或者SIP URI,用来表示请求要送往的服务或用户信息,其中不能包括控制字符,也不能包含在"<>"之中.
* **SIP-Version**: SIP的版本号,与RFC3261对应的是"SIP/2.0".和HTTP/1.1的处理类似,但不同点为SIP处理版本号是以字符串的格式,虽然这在实践中并没什么太大关系.

1. **Status-Line = SIP-Version SP Status-Code SP Reason-Phrase CRLF**

状态码(Status-Code)由3位数字组成,表示请求的结果. 状态码的第一位表示响应的种类:

* 1xx(表示100-199,下同): 临时响应(Provisional),表示请求已经被收到,但还在处理之中.
* 2xx: 成功(Success), 请求被成功接收,理解以及被接受.
* 3xx: 重定向(Redirection), 可能需要重新选择发送地址以完成请求.
* 4xx: 客户端错误(Client Error), 请求包含错误的语法,或者不能被服务器完成.
* 5xx: 服务端错误(Server Error), 服务器处理一个合法的请求失败.
* 6xx: 失败(Global Failure),

消息头结构：field-name: field-value，其中冒号前面没有空格，冒号后面一个空格

field-name可以是如下内容：

* Via: 包含了发送方想接受此次请求对应响应的地址,这里是pc33.atlanta.com,  
  并且还包含了识别此次传输事务的分支参数(branch parameter).实际是指定了请求传递的路径
* Max-Forwards: 用来限制请求的最大跳数(max hops),在每个hop之后递减少.
* To: 包含了此次请求的目的用户的显示姓名Bob(display name)以及SIP/SIPS URI(sip:bob@biloxi.com)
* From: 包含了此次请求的发送方的显示姓名Alice和URI, 除此之外还有一个tag参数,包含了随即的字符串,  
  将用于添加在URI中,主要用于验证和区分.
* Call-ID: 包含了此次通话中全局不同的标识,由随机字符串和发送端的主机名或IP地址组合而成.To,From和Call-ID  
  字段完全定义了一个Alice和Bob的端到端的SIP关系,并表示为当前对话(dialog).
* CSeq: 或者写为Command Sequence,包含了一个整数(CSeq号)和方法名,CSeq号在本次对话中随着每次新的请求而递增.
* Contact: 包含代表直接连接Alice的SIP/SIPS URI. 和Via段不同的是,Via告诉其他单位要往那发送响应,  
  而Contact告诉其他单位要往哪发(以后的)请求.
* Content-Length: 消息体的长度.
* Content-Type: 消息体(message body)的格式, 如SDP信息则为"application/sdp",关于SDP可以参考前一篇博客[P2P通信标准协议(三)之ICE](http://pannzh.github.io/tech/p2p/2015/12/20/p2p-standard-protocol-ice.html).