

# SIP 协议简介

## (一) 之 SIP 基本的概述

补充说明：RFC 即 Request For Comments（RFC），是一系列以编号排定的文件。文件收集了有关互联网相关信息，以及 UNIX 和互联网社区的软件文件。目前 RFC 文件是由 Internet Society（ISOC）赞助发行。基本的互联网通信协议都有在 RFC 文件内详细说明。RFC 文件还额外加入许多的论题在标准内，例如对于互联网新开发的协议及发展中所有的记录。因此几乎所有的互联网标准都有收录在 RFC 文件之中。（摘自百度百科）

在这里提供给大家一个下载相关 RFC 文档的链接：<https://www.rfc-editor.org/>，在这里你可以找到相关的原始文档。

## 1、SIP 协议介绍

Internet 的许多应用都需要建立和管理一个会话，会话在这里的含义是在参与者之间的数据的交换。由于考虑到参与者的实际情况，这些应用的实现往往是很复杂的：参与者可能是在代理间移动，他们可能可以有多个名字，他们中间的通讯可能是基于不同的媒介（比如文本，多媒体，视频，音频等）—有时候是多种媒介一起交互。人们创造了无数种通讯协议应用于实时的多媒体会话数据比如声音，影像，或者文本。本 SIP（会话初始协议）和这些协议一样，同样允许使用 Internet 端点（用户代理）来寻找参与者并且允许建立一个可共享的会话描述。为了能够定位精确的会话参与者，并且也为了其他的目的，

SIP 允许创建基础的 **network hosts**（叫做代理服务器），并且允许终端用户注册上去，发出会话邀请，或者发出其他请求。SIP 是一个轻形的，多用途的工具，可以用来创建，修改和终止会话，它独立运作于通讯协议之下，并且不依赖建立的会话类型。

## 2、SIP 协议功能概况

SIP 是一个应用层的控制协议，可以用来建立、修改、和终止多媒体会话（或者会议）例如 Internet 电话。SIP 也可以邀请参与者参加已经存在的会话，比如多方会议。媒体可以在一个已经存在的会话中方便的增加（或者删除）。SIP 显示的支持名字映射和重定向服务，这个用于支持个人移动业务—用户可以使用一个唯一的外部标志而不用关系他们的实际网络地点。SIP 在建立和维持终止多媒体会话协议上，支持 5 个方面：

- 1) 用户定位： 检查终端用户的位置，用于通讯。
- 2) 用户有效性： 检查用户参与会话的意愿程度。
- 3) 用户能力： 检查媒体和媒体的参数。
- 4) 建立会话：“ringing”,建立会话参数在呼叫方和被叫方。
- 5) 会话管理： 包括发送和终止会话，修改会话参数，激活服务等  
等。

SIP 不是一个垂直集成的通讯系统。SIP 可能叫做是一个部件更合适，它可以用作其他 IETF 协议的一个部分，用来构造完整的多媒体架构。比如，这些架构将会包含实时数据传输协议（RTP）（RFC 1889）用

来传输实时的数据并且提供 QoS 反馈，实时流协议（RSTP）(RFC 2326)用于控制流媒体的传输，媒体网关控制协议（MEGACO）(RFC 3015)用来控制到公共电话交换网（PSTN）的网关，还有会话描述协议（SDP）（RFC 2327）用于描述多媒体会话。因此，SIP 应该和其他的协议一起工作，才能提供完整的对终端用户的服务。虽然基本的 SIP 协议的功能组件并不依赖于这些协议。

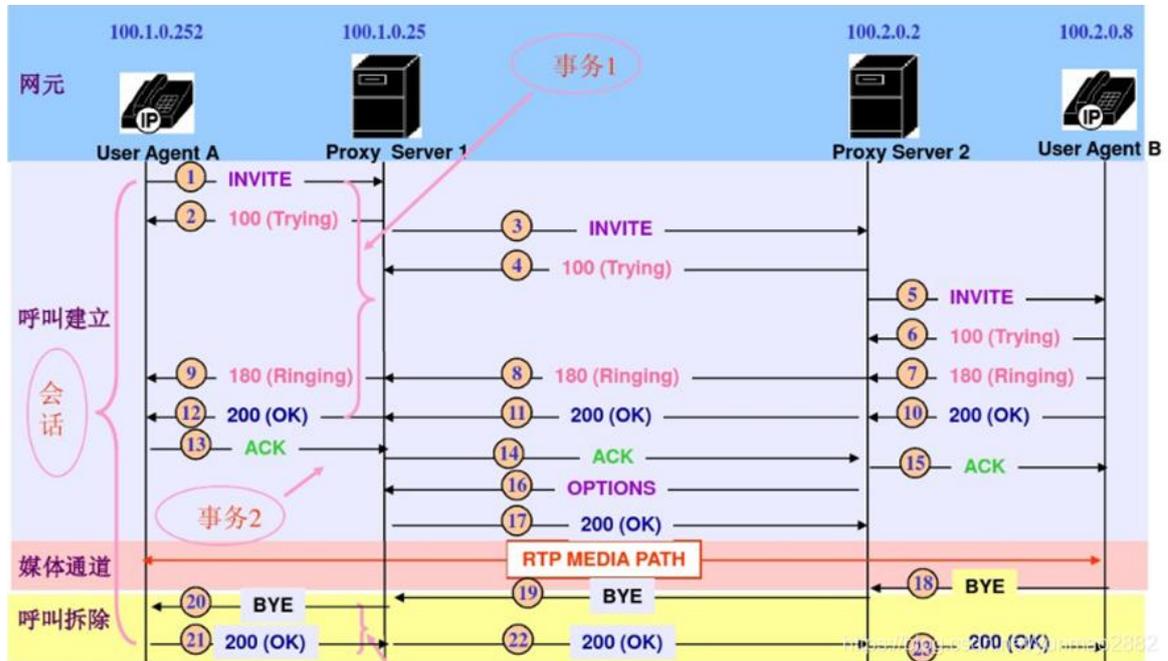
SIP 本身并不提供服务。但是，SIP 提供了一个基础，可以用来实现不同的服务。比如，SIP 可以定位用户和传输一个封装好的对象到对方的当前位置。并且如果我们利用这点来通过 SDP 传输会话的描述，立刻，对方的用户代理可以得到这个会话的参数。如果我们用这个像传输会话描述（SESSION DESCRIPTION SD）一样呼叫方的照片，一个“呼叫 ID”服务很容易就建立了。这个简单的例子说明了，SIP 作为一个基础，可以在其上提供很多不同的服务。

SIP 并不提供会议控制服务（比如议席控制或者投票系统），并且并没有建议会议应该怎样管理。可以通过在 SIP 上建立其他的会议控制协议来发起一个会议。由于 SIP 可以管理参与会议的各方的会话，所以会议可以跨异构的网络，SIP 并不能，也不打算提供任何形式的网络资源预留管理。

安全对于提供的服务来说特别重要。要达到理想的安全程度，SIP 提供了一套安全服务，包括防止拒绝服务，认证服务（用户到用户，代理到用户），完整性保证，加密和隐私服务。

SIP 可以基于 IPV4 也可以基于 IPV6。

### 3、基于 SIP 协议会话建立的基本过程（图解）



上图为基于 SIP 的会话发起的基本过程。用 SIP 建立通讯通常需要六个步骤：

- 1) 注册，发起和定位用户
- 2) 进行媒体协商—通常采用 SDP 协议来携带媒体参数
- 3) 由被叫方来决定是否接纳该呼叫
- 4) 呼叫媒体流建立并交互
- 5) 呼叫更改或处理如呼叫转移等
- 6) 呼叫终止

SIP 是基于一个类似 HTTP 协议的请求应答的通讯模式。每一个通讯都包含对某个功能的请求，并且起码需要一个应答。在这个应答中，通信双方的 Alice 的软电话发送一个含有通信双方的 Bob 的 SIP URI

抵制的 INVITE 通讯请求。INVITE 是一个 SIP 请求的例子，表示请求方（Alice）希望服务方（Bob）应答。INVITE 请求包含一系列的包头域（Header fields）。包头中包含很多属性并且包含了传输消息的附加信息。在 INVITE 中有如下的字段：呼叫的唯一标志、目的地址、Alice 的地址、Alice 和 Bob 建立会话的类型。

在前面内容中，通过图片的形式描述了一个会话发起的过程，在会话发起时，主叫方会发送 INVITE 信令来发起建立会话的请求，这个信令的基本内容如下：

```
INVITE sip:20151007002@20.0.0.113:5060 SIP/2.0
Max-Forwards: 50
Via: SIP/2.0/UDP 20.0.0.2:5060;rport;branch=z9hG4bK624077667
From: <sip:1000@20.0.0.2>;tag=1299473648
To: <sip:20151007002@20.0.0.113:5060>
Call-ID: 785496241@20.0.0.2
CSeq: 79 INVITE
User-Agent: ZPZZ
Contact: <sip:1000@20.0.0.2:5060>
Allow: ACK, INVITE, BYE, CANCEL, MESSAGE, REGISTER, REFER,
OPTIONS, INFO, UPDATE
Supported: timer
Content-Type: application/sdp
Content-Length: 465

v=0
o=ZPZZ 1444707847 1444707847 IN IP4 20.0.0.2
s=Play
c=IN IP4 10.0.0.2
b=AS:4096
t=0 0
a=rtpport-mux
a=muxid:222222
m=audio 1234 RTP/AVP 8 101
c=IN IP4 10.0.0.2
a=rtpmap:8 PCMA/8000
a=ptime: 20
a=rtpmap:101 telephone-event/8000
a=rtpid:1025151
a=recvonly
m=video 1234 RTP/AVP 96
b=TIAS:4032000
c=IN IP4 10.0.0.2
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=640020;max-mps=243000;max-fs=8100;
a=rtcp_fb:ccm
a=rtpid:1025858
a=recvonly
```

<https://blog.csdn.net/yunmao2882>

往往会结合 SDP 这个会话描述协议来描述整个媒体的会话参数，

比如说音视频的编码格式、音视频的编码码率、音视频的采样率、视频的分辨率等等。这个可以通过 SIP 的消息正文来描述，可以通过其他定义的协议在正文中进行描述。有一种是会话描述协议（Session Description Protocol SDP）（RFC2327[1]）。这个 SDP 消息通过 SIP 的消息中传送，就像通过附件发送 EMAIL 一样，或者说通过 HTTP 传输的网页一样。

在文本消息的第一行，包含了请求的类型（INVITE）。在这行之后的是这个请求的头域。这个例子中包含了最少需要的头域集合。简单介绍一下：

**VIA** 域包含了 Alice 接收发送请求的服务器地址

（[pc33.atlanta.com](http://pc33.atlanta.com)）。同样这个包含了一个分支参数来标志 Alice 和这个服务器的会话事务。

**TO** 域包含了显示姓名（Bob）和一个 SIP 或者 SIPS URI（sip:bob@biloxi.com）请求将首先传输到这个 URI 中。显示姓名（Display names）在 RFC 2822 中描述。

**From** 域也同样包含一个显示姓名（Alice）和一个 SIP 或者 SIPS URI

（sip:alice@atlanta.com）这个 URI 用来标志请求的原始发起者。

这个域也包含了一个 TAG 参数，这个 TAG 参数是一个随机字符串

（1928301774），是软电话（softphone）在 URI 上增加的一个随机串。用来做标志用途的。

**Call\_ID** 包含一个全局的唯一标志，用来唯一标志这个呼叫，通过随机字符串和 softphone 的自己名字或者 IP 抵制混和产生的。通过 TO TAG,

**FROM TAG** 和 **CALL-ID** 完整定义了 **Alice** 和 **Bob** 之间的端到端的 SIP 关系，并且表示这个是一个对话性质的关系。

**CSEQ** 或者 **Command Sequence** 包含了一个整数和一个请求名字。

这个 **Cseq** 数字是顺序递增的。每当对话中发起一个新的请求都会引起这个数字的顺序递增。

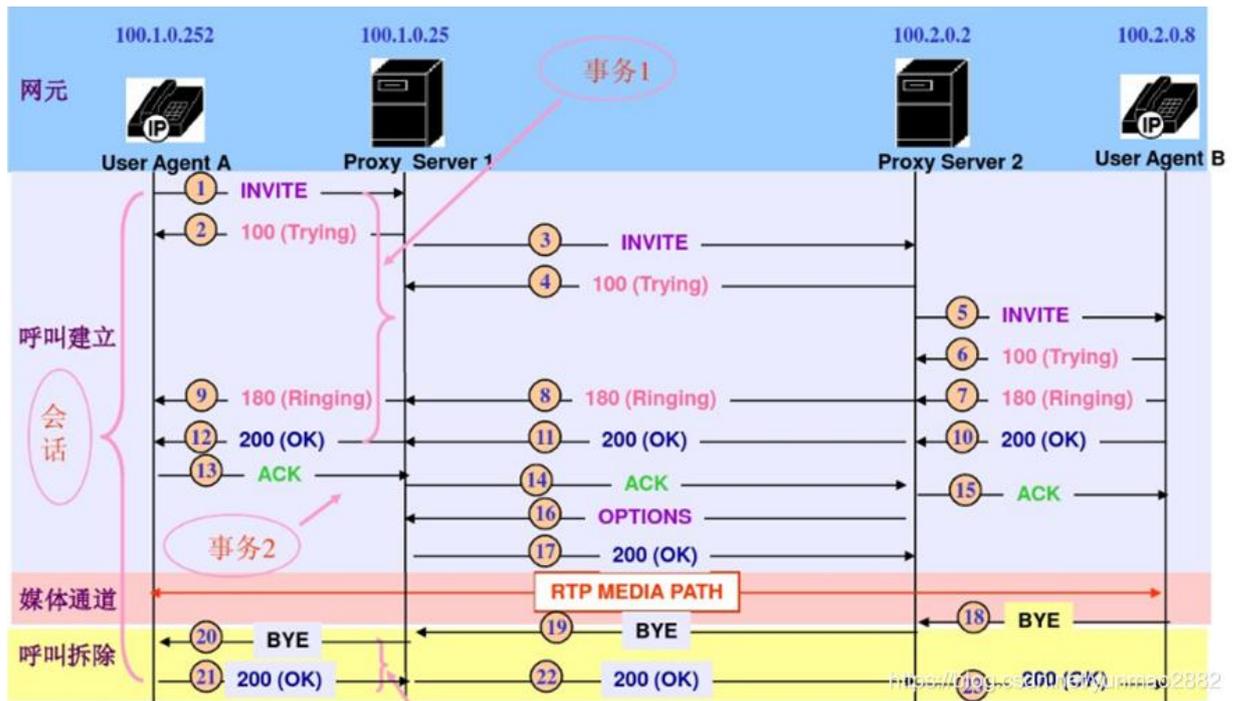
**Contact** 域包含一个 SIP 或者 SIPS URI 用来表示访问 **Alice** 的直接方式，通常由用户名和一个主机的全名（**Fully Qualified Domain Name FQDN**）组成。当 **FQDN** 作为首选的时候，许多终端用户由于不会由名字登记（而导致不能访问 **Alice** 的主机），所以 **IP** 地址是可选的。

**VIA** 域告诉大家本请求发送到哪里并且应答到哪里，**Contact** 域告诉大家将来的请求将发送到哪里（奇怪...不是 **Alice** 发起的么，将来的请求应该是 **Bob** 才对啊）。

**Max-Forwards**:最大转发数量限制了通讯中转发的数量。它是由一个整数组成，每转发一次，整数减一。

**Content-type** 包含了消息正文的描述（比如前面提到的 `application/sdp`）

**Content-length**:包含消息正文的长度（字节数）



这个也是任何一个信令中不可或缺的内容。由于 softphone 并不知道 bob 或者 bob 的 sip 服务器 biloxi.com 在哪里，所以 softphone 发送 INVITE 请求到 Alice 的 sip 服务器,atlanta.com。这个 atlanta.com SIP 服务器应该已经在 Alice 的 softphone 中配置了，或者可以通过 DHCP 获得。atlanta.com SIP 服务器是一台代理服务器。代理服务器接收 SIP 请求并且根据请求转发。在这个例子中，代理服务器接收到 INVITE 请求，并且回送一个 100 (Trying) 应答给 Alice 的 softphone。100 (Trying) 应答表示 INVITE 请求已经收到，并且代理服务器正在转发 INVITE 请求。

SIP 的应答是通过一个三位数的数字表示的。SIP 应答同样包含 TO、FROM、Call-ID, CSEQ 和在 VIA 中的分支参数，这个参数使得 Alice 的 softphone 可以把请求和应答关联起来。atlanta.com 代理服务器收到 INVITE 请求之后，就去找 biloxi.com 可能通过 DNS 服务来找提供这个 biloxi.com 的 SIP 服务器。最后，转发 INVITE 请求到

biloxi.com 或者能到达 biloxi.com 的代理服务器。在转发请求之前，atlanta.com 代理服务器会在 via 头上增加一个一段包含自己抵制的值（INVITE 已经包含了 Alice 的地址 VIA 域）。biloxi.com 代理服务器收到这个 INVITE 请求并且返回一个 100（Trying）应答给 atlanta.com 代理服务器标志这它已经收到这个请求并且正在处理这个请求。这个代理服务器通过查询数据库，通常叫做地址服务，这个服务中包含了 bob 的当前 ip 地址。biloxi.com 代理服务增加另一段包含自己地址的 VIA 头域并且发送它到 bob 的 sip 电话。

Bob 的 SIP 电话接收到 INVITE 请求并且提醒 bob 有一个从 Alice 的呼入，这样 bob 可以决定是否响应这个呼入。这个意思就是：bob 的电话响了。bob 的 sip 电话发送一个 180（Ringing）回应，这个回应将通过两个代理服务器原路返回给 Alice。每一个代理服务器通过 via 头域决定该把这个应答发送给哪里，并且在发送之前把自己的地址从头上拿走。虽然 DNS 和定位服务在路由最初的 INVITE 请求，180（ringing）响应可以简单返回给发起者而不需要查找发起者在哪里，并且不需要在代理服务器保留状态，同时，每一个转发 INVITE 的代理也可以得到 INVITE 的每一个应答，这样的特性也非常有用。

当 Alice 的 softphone 收到 180（Ringing）应答的时候，它提示 Alice,可能是通过一个回铃音，或者屏幕上的一个消息提示。Bob 决定响应这个呼叫。当他拿起电话，他的 SIP 电话发送 200（OK）回应给发送者，表示这个电话已经接起来了。这个 200（OK）包含了一个消息体，这个消息体包含 SDP 媒体描述，这个媒体描述包含 Bob 希望

和 Alice 建立何种媒体连接。同样，SDP 消息也是两段交换：Alice 发送一个给 Bob，Bob 发送一个回给 Alice。这个两段的交换提供基本的兼容性协商，并且基于简单的 SDP 提出/应答交换模型。如果 Bob 不想响应这个呼叫或者正在响应别的呼叫，一个错误的响应会代替正常的 200 (OK) 回送出去，这样，就不会有连接建立。

当正常情况发生时，200 (OK) 应答通过两个代理并且发送到 Alice 的 softphone 上，Alice 的 softphone 收到这个应答，停止振铃，并且标志电话 Bob 已经接听。最后，Alice 的电话发送一个确认消息，ACK，到 Bob 的 SIP 电话来确认接收到了这个最后的 200 (ok) 应答。在这个例子中，ACK 信号是直接由 Alice 的 softphone 发送到 Bob 的 SIP phone 上，跨过了两个代理服务器。这是因为两个端点 (Alice 和 Bob) 通过 INVITE/200(OK) 的请求应答包中的 Contact 包头域都知道互相之间的地址了，这个地址是最开始发起 INVITE 请求的时候所不知道的。所以，不需要两个代理服务器再查找对方的地址了，所以代理服务器不参与接下来的通话流了。这就完成了一个完整的使用 INVITE/200/ACK 三方握手来建立 SIP 会话的过程。会

现在，Alice 和 Bob 的媒体会话开始了，他们通过发送刚才建立会话所交换的 SDP 包中约定的互相明白的媒体包来进行会话。一般情况下，端到端的媒体包和 SIP 信号控制包通过不同的通讯路径来发送。**注意：**在这个过程中，其实通信的双方就可以进行媒体会话层的协商，比如具体整个通信双方采用什么样的音视频格式，Alice 端采用什么格式来进行编码，Bob 端采用什么类型解码等，这个都可以在这个

协商过程中敲定下来，具体采用什么样的机制，各自根据自己的需求进行设计即可。在进行代码实现的时候，可以去找 **exosip** 开源库，同时在媒体层进行会话参数协商时，自行设计实现对相关信令的解析。