1. sip协议应用场景举例；

Internet多媒体会议 、IP电话或多媒体分发。例如，一个人可以使用SIP发起对另一个人的电话呼叫，或者有人可以与许多参与者建立电话会议。

广泛应用于电路交换、下一代网络以及IP多媒体子系统的网络中，可以支持并应用于语音、视频、数据等多媒体业务，同时也可以应用于Presence（呈现）、Instant Message（即时消息）等特色业务。可以说，有IP网络的地方就有SIP协议的存在。

SIP 能够连接使用任何 IP 网络和任何 IP 设备的用户，改进了企业和用户的通信方式。基于 SIP 的应用（如 VOIP、多媒体会议、push-to-talk（按键通话）、定位服务、在线信息和 IM）即使单独使用，也会为服务提供商、ISV、网络设备供应商和开发商提供许多新的商机。不过，SIP 的根本价值在于它能够将这些功能组合起来，形成各种更大规模的无缝通信服务。

1. 列举出sip协议相关标准文档以及主要内容描述；

SDP（会话描述协议）rfc2327 ：用于描述多媒体会话

RTP（实时传输协议）rfc1889 ：用来传输实时的数据并且提供QoS反馈

RSTP（实时流协议）rfc2326 ：用于控制流媒体的传输

MEGACO（媒体网关控制协议）rfc3015 ：用来控制到公共电话交换网（PSTN）的网关

LDAP (轻型目录访问协议): 负责定位

RADIUS（远程用户拨号认证系统）：负责身份验证的远程身份验证拨入用户服务

1. sip协议消息类型和必须头字段描述；

SIP的结构与HTTP（客户－服务器协议）相似（基于IP协议，并利用IP网络），体现了客户端 - 服务器体系结构，是一个Client/Sever协议，因此SIP消息有两种类型 - 请求和响应。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。消息都包含头字段和消息字段。

请求和应答中必须头域：

VIA：用来描述请求当前经历的路径的，并且标志了应答所应当经过的路径。Via头域的branch ID参数提供了事务的标志，并且用于proxy来检查循环路由。Via头域包含了用于发送消息的通讯协议，客户端主机名或者网络地址，可能还有接收应答所用的端口号码。Via头域还可以包含参数“maddr”,“ttl”,“received”和“branch”, 对于遵循本规范的实现，这个branch参数的值必须用magic cookie“z9hG4bK”打头

\*To：定义了逻辑上请求的接收者。选项“display-name”意味着展示给客户的界面。“tag”参数提供了对话识别机制。

\*From: 表示了请求的来源地。这个可能和对话的来源的不同，被叫方到呼叫方的请求会在From头域使用被叫方的地址。

\*Call\_ID: 包含一个全局的唯一标志，用来唯一标志这个呼叫，通过随机字串和softphone的自己名字或者IP地址混和产生的。

UAC发出的给某个注册服务器的所有注册请求都应该有相同的Call-ID头域值。如果相同的客户端使用了不同的Call-ID值，注册服务器就不能察觉是否延迟的还是来的时候发生故障了。

\*CSeq: 请求中的Cseq头域包含了一个单个的数字序列号和请求的方法。这个序列号必须是表示成为一个32位的无符号整数。在Cseq的请求方法部分是大小写敏感的。Cseq头域是为了在会话中对事务进行排序的，提供事务的唯一标志，并且区分请求和请求的重发。如果序列号相等，并且请求的方法相等，那么两个Cseq头域就是相等的。

\*Contact：头域提供了一个URI，这个URI的含义取决于是在请求还是在应答中。

4、sip注册的目的以及相关信令；

5、sip Register消息中，From/To/contact和Expires字段含义；（10.2）

REGISTER头域必须包含Request-URI、To、From、Call-ID、CSeq、Contact等头域。

Request-URI：表达了登记服务所指明的位置服务所在的区域。

From：包含了可靠的注册的人员的记录地址address-of-record。

Call-ID：UAC发出的给某个注册服务器的所有注册请求都应该有相同的Call-ID头域值。如果相同的客户端使用了不同的Call-ID值，注册服务器就不能察觉是否延迟的还是来的时候发生故障了。

CSeq：保证了Register请求的正确顺序。一个UA为每一个具备相同的Call-ID的Register请求顺序递增这个CSeq值。

To：这个头域包含了注册被查询、增加、修改的address-of-record。

Contact：可选的头域。这个头域可以有0个或者多个包含绑定地址信息的值。UAS在没有收到上一个注册的应答或者上一个REGISTER请求超时之前,禁止发送新的注册请求(就是说.包含一个新的Contact头域值，而不是重发)。

Expires：表示这个注册的地址的有效期。

Contact头域参数expires：表明UA希望的绑定的有效时间. 以秒为单位。如果未提供此参数，则使用Expires头域值的值。

【(第八节) 在Contact头域中的“expires”参数指明了这个URI的生存周期。缺省3600】

6、如何区分sip的注册/保活/注销；

7、sip注册摘要认证的过程，是否是必须的；

8、如何创建一个sip会话，什么是sip“早期”会话，如何更新一个会话；

当一个会话由INVITE建立的时候，每一个由不同UAS的1xx或者2xx的应答创建一个对话，并且当完成了会话描述的请求/应答（offer/answer）交互之后，它也就创建了一个会话。

Sip会话是通过使用特定方法生成对请求的没有失败的响应来创建的。在这个规范中，只有包含To tag的2xx和101-199应答，并且请求是INVITE的，会建立一个对话。

当收到一个非终结应答的（临时应答）时候，对话会建立成“early”状态，并且成为一个早期会话。

在现存的碎花中发出re-INVITE请求

单个的re-INVITE请求可以同时更改对话和会话的参数。呼叫方或者被叫方都可以更改现存的会话。

9、如何结束一个会话；

对于INVITE初始化的对话，终止对话需要发送一个BYE请求。

如果初始化的INVITE产生了非2xx的终结应答，它也终结了由本次请求创建的任何会话（如果有的话），并且终结了所有的本次请求创建的对话（如果有的话）。

BYE请求用于终结指定的会话或者尝试建立的会话。当在对话中接收到了一个BYE,任何与该对话相关的会话都应当终止。

在建立对话中的终结对话，跟请求方法无关，如果对话外的请求产生了一个非2xx终结应答，任何前边请求创建的“早期对话”(early dialogs)将会终止。在已经建立的对话中，终结对话就是请求方法相关的。在这个定义中，BYE方法将会终结一个对话。15节有细致的讨论。

10、sip中的offer-answer交互的相关信令和规则，SDP中的通道协商规则；

11、invite/200 ok中的Request-URI，record-route, route，contact的生成规则，以及后续请求相关字段的生成规则；

12、sip proxy的收到请求和响应的相关处理；

proxy是一个SIP逻辑上的概念。当接收到一个请求，在做代理服务器之前，首先应该有一个部件来决定是否自身需要响应这个请求。

proxy对于每一个新的请求来说，既可以作为有状态的也可以作为无状态的模式来处理。当作为无状态的处理模式的时候，proxy就是简单的转发。它转发每一个请求下行到一个由请求所决定的目的地。并且简单转发从上行流取得的应答。一个无状态的proxy在处理完一个消息之后就会丢弃这个消息的相关资料。有状态的proxy会保留这些信息（尤其是事务信息），保留每一个接收的请求和每一个接收请求的应答的相关信息。它保留这些信息用于处理与这个请求相关的后续消息。一个有状态的proxy可以选择“分支”一个请求，路由它到多个地点。任何被路由到多个地点的请求都必须当作有状态的处理。

对于每一个目的地，proxy转发请求都遵循下列步骤：

1、拷贝一个接收到的请求

2、更新Request-URI

3、更新Max-Forwards头域

4、可选增加一个Record-Route头域

5、可选增加附加的头域

6、路由信息后处理

7、决定下一个节点地址、端口、通讯协议。

8、增加一个Via头域值

9、如果需要，增加一个Content-Length头域

10、 转发这个新的请求

11、设置定时器C

当proxy收到一个应答的时候，它首先尝试定位一个与这个应答匹配的客户端事务（17.1.3）。如果没有匹配，proxy必须作为无状态的proxy来处理这个应答（即使这个应答是信息性质的应答）。如果与应答匹配的客户端事务找到了，那么这个应答将转给这个客户端事务进行处理。

将应答转给对应的客户端事务（或者更通常的说法是发出请求的或者相关的事务），并不是为了更强大的处理能力，它是保证了“晚到”的给INVITE请求的2xx应答能够正确的转发。

当客户端事务把应答交给proxy层，将会执行下列步骤：

1、寻找适当的应答上下文

2、用临时应答来更新定时器C

3、从最上边移除Via

4、在应答上下文中增加应答

5、检查这个应答是否需要立刻发送

6、如果需要，从应答上下文中选择最好的终结应答。

如果在与这个应答上下文相关的每一个客户端事务都结束的以后，还是没有终结应答转发，那么proxy必须选择从已经收到的应答中，选择转发“best”应答回去。

下列步骤必须在每一个被转发的应答上执行。就像每一个请求有超过一个应答被转发一样：至少有一个终结应答和0个或者多个临时应答。

7、需要合并认证头域值。

8、 可选的重写Record-Route头域值

9、转发应答

10、产生合适的CANCEL请求。

**（第1节）Introduction**

**（第2节）Overview of SIP Functionality**

**（第3节）**

**（第4节）**

**（第5节）**

**（第6节）**

**（第7节）**

**（第8节）**

**（第9节）取消一个请求 Cancel**

CANCEL一般取消invite请求

需要与被取消的请求的Request-URI、Call\_ID、To、CSeq的数值部分、From这些头域及头域的tag相同

CANCEL请求是基于“点对点”（hop-by-hop）的，也是不能再提交的。

CANCEL请求不能包含Require和Proxy-Require头域的

**（第10节）注册 Registrations**

注册服务为特定地区的位置服务创建绑定关系，这个绑定关系是用来建立包含一个或者多个联系地址的address-of-record URI。

Register请求可以用来增加、删除、查询绑定资料。

**（第11节）查询能力 OPTIONS**

SIP方法OPTIONS允许一个UA来查询另外一个UA或者proxy服务器的能力。

**（第12节）会话 Dialog**

对话是表现为两个用户代理（UA）之间的持续一段时间的点对点的SIP关系。

**（第18节）通讯 Transport**

通讯层负责请求和应答在网络上的实际传输。这包括了在面向连接的通讯方式下的请求和应答所使用的连接管理。

通讯层负责管理像TCP/SCTP之类通讯协议的长连接，或者在这些协议上的TLS连接，并且包括管理打开这些连接的使用者的管理。这包括了客户端或者服务端通讯层打开的连接，这样在客户端服务端通讯函数可以共享这些连接。

我们建议在实现中，当发送（或者接收）完成最后一个消息之后，依旧维持这个连接一段时间（这段时间可以是实现自己定义的时间）。这段时间应当是至少等于本节点的事务从创建到结束的最长时间。这是为了让事务能够在他们所创建的同一个连接上完成（比如，在这个连接上完成请求，应答的处理，在INVITE的情况下的给非2xx的ACK应答等等）。这通常意味着至少64×T1秒。

所有的SIP元素都必须实现基于UDP和TCP的通讯。SIP元素还可以实现其他的协议。

通讯层的客户端负责发送请求和接收应答。通讯层的用户把请求交给通讯层的实例进行处理，包括IP地址端口，通讯层实例，还有可能有多点广播的TTL。

一个服务器应当能够接收从任何IP地址、端口和协议上过来的请求。他们是通过对这个服务器的SIP或者SIPS URI（附件[4]）的DNS查找，得到这个服务器的地址然后连接和发送的请求的。

**（第19节）**

Option tags是一个唯一标志，用来指明SIP中的新options（扩展）的。

“tag”参数用于SIP消息中的To和From头域。它作为一个通用的机制的一部分来唯一标志一个对话，这个机制用Call-ID和两个从对话参与者的tag来标志一个对话。当UA在对话外发出一个请求时，它只包含了From tag,提供了对话ID的“一半”。对话根据应答创建完成，这个应答在To头域中提供了对话ID的另一半。SIP请求的分支意味着一个单个请求可以创建多个对话。这个也解释了为何需要对话两方的标志；如果没有被叫方的标志，呼叫方不能分辩和消除由单个请求创建的多个对话。

**（第20节） 头域 Header Filelds**

Accept头域的语法定义遵从[H14.1]。语义也相同，但如果不存在接受头字段，则服务器应假定默认值application/sdp。

空的Accept头域意味着不接受任何格式。

Accept-Encoding头域类似Accept，但是限定了接收应答中的内容的编码[H3.5]。

一个空的Accept-Encoding头域是允许的。他等同于Accept-Encoding: identity,这就是说，只有identity编码，也就是说没有编码的情况，是允许的。

如果没有Accept-Encoding头域存在，那么服务端应当使用缺省值：identity。

Accept-Language头域用来在请求中指定首选的语言的，这个首选的语言是在应答中的消息体中的原因分析，会话描述，或者状态报告的。如果没有Accept-Language存在，那么服务端应当假设所有的语言客户端都可以接受。

当INVITE请求有一个Alert-Info头域的时候，Alert-Info头域就包含的是给UAS的一个额外的信息。

Allow头域列出了UA支持的方法列表。

Authentication-Info 头域提供了和HTTP类别相同的认证方法。UAS可以在给一个顺利通过认证的请求的2xx应答中包含这个头域，并且是使用基于Authorization头域的分类。

Authorization头域包含了了UA进行认证的信任书。

Call-ID头域用来唯一区别一个特定的邀请或者一个特定客户端的所有注册项。

Call-Info头域提供了对呼叫方或者被叫方的附加信息，如果出现在请求中则是呼叫方的信息，如果出现在应答中则是被叫方的。

Contact头域提供了一个URI，这个URI的含义取决于是在请求还是在应答中。

本文档定义了一个Contact参数“q”和“expires”。这些参数只有当Contact头域在REGISTER的请求或者应答，或者3xx的应答中才有效。

Content-Disposition头域描述了消息体，或者消息的多个部分，或者消息体的一个部分应被UAC或者UAS怎样解释。这个SIP头域扩展了MIME Content-Type。

Content-Encoding头域是对”media-type”(媒体类型)的一个修正。当存在这个头域的时候，它的值就是对包体内容编码的附加说明，并且因此必须根据本字段应用正确的解码机制，这样才能得到正确的Content-Type头域指出的媒体类型的解码。Content-Encoding首要应用于在不丢失媒体类型标记的情况下对消息体进行压缩处理。

Content-Language

Content-Length头域标志了消息体的大小，给消息的接受者，以10进制表示的数字。应用程序应当使用这个字段标志的大小来传送消息体，而不关心消息体的媒体类型是什么。如果是基于流的通讯协议（比如TCP），那么本头域必须提供。

Content-Type头域标志了发给对方的消息体的媒体类型。

请求中的Cseq头域包含了一个单个的数字序列号和请求的方法。这个序列号必须是表示成为一个32位的无符号整数。在Cseq的请求方法部分是大小写敏感的。Cseq头域是为了在会话中对事务进行排序的，提供事务的唯一标志，并且区分请求和请求的重发。如果序列号相等，并且请求的方法相等，那么两个Cseq头域就是相等的。

Date头域包含了日期和时间。

Error-Info头域提供了对有错误应答码的应答的附加信息。

Expires头域给定了消息（或者内容）过期的相关时间。这个字段的精确定义是方法相关的。对于一个INVITE的超时时间并不影响这个INVITE请求建立的实际的会话。不过，会话描述协议可以描述在一个会话上的的时间限制。

From头域表示了请求的来源地。这个可能和对话的来源的不同，被叫方到呼叫方的请求会在From头域使用被叫方的地址。

In-Reply-To头域列举了本次呼叫相关的或者返回的Call-ID。这些Call-ID可以被客户端cache起来，这样可以在这个头域中返回。

Max-Forwards头域必须在任何一个SIP请求中使用，来限制中间转发请求到下一个节点的proxy或者gateway的个数。这个在客户端trace一个请求，如果路由失败或者在中间出现循环的时候特别有用。

Max-Forwards是一个0－255的整数，表明了在这个请求消息中允许被转发的剩余次数。每当服务器转发这个请求一次，这个数字就减一。建议的初始值是70。

Min-Expires头域包含了一个服务器所支持的内部状态（soft－state）的最小的刷新时间间隔。这包括由注册器存储的Contact头域。

MIME-Version

Organization头域包含了发出请求或者应答的SIP节点所属的组织名字。这个字段可以用来让客户端软件过滤呼叫。

Priority头域标志了客户端评价的请求紧急程度。Priority头域描述了SIP应当处理人工或者UA发过来的请求的优先级。

Proxy-Authenticate头域用来进行认证使用的。

Proxy-Authorization头域允许客户端向一个要求认证的proxy证明自己（或者证明它的使用者）的身份。一个Proxy-Authorization头域包含了与UA认证信息相关的信任书，这个信任书是给proxy和/或者本请求相关的域的。

Proxy-Require头域用来表示请求中一定要求proxy支持的相关的特性。

Record-Route头域是proxy在请求中增加的，用来强制会话中的后续请求经过本proxy的。

Reply-To头域包含了逻辑上返回目的地URI，这个可以和From头域不同。比如，URI可以用来返回未接电话或者未建立的会话。如果用户希望保留匿名，那么这个头域应当从请求中去除或者改变，这样可以避免透露个人隐私信息。

Require头域用于UAC告诉UAS关于要求UAS支持那些特性。虽然这是一个可选的头域，但是如果Require头域存在，那就一定不能掠过不处理。

头域包含一个option tag的列表。每一个option tag定了一个要处理请求要求UAS必须支持的SIP扩展。通常，这用于定义一个需要支持的扩展头域的集合。

Retry-After头域可以用于500（Server Internal Error）或者503（Service Unavailable）应答，用来标志大约本服务还会处于不可用状态多久。在404(Not Found)，413(Request Entity Too Large)， 480(Temporarily Unavailable)，486(Busy Here)， 600 (Busy)， 或者603(Decline)应答中用于标志何时被叫方会恢复正常。

Route头域用于强制一个请求经过一个proxy路由列表。

Server头域包含了关于UAS处理请求所使用的软件信息。

Subject头域提供了呼叫的一个概览，允许呼叫不用分析会话描述就可以大致过滤。会话描述并不需要和INVITE邀请使用相同的主题标志。

Supported头域列举了UAC或者UAS支持的扩展。 Supported头域包含了一个option tag的列表，在19.2节描述的option tag,他们是这个UAS或者UAC所支持的。遵循本规范的UA必须只包含遵循标准RFC扩展的option tag。如果本字段是空的，意味着不支持任何扩展。

Timestamp头域描述了当UAC发送请求到UAS的时间戳。

To头域定义了逻辑上请求的接收者。选项”display-name”意味着展示给客户的界面。”tag”参数提供了对话识别机制。

Unsupported头域列出了不被UAS支持的特性列表。

User-Agent头域包含了发起请求的UAC信息。

Via头域是用来描述请求当前经历的路径的，并且标志了应答所应当经过的路径。Via头域的branch ID参数提供了事务的标志，并且用于proxy来检查循环路由。

Via头域包含了用于发送消息的通讯协议，客户端主机名或者网络地址，可能还有接收应答所用的端口号码。Via头域还可以包含参数“maddr”,“ttl”,“received”和“branch”,这些定义在其他节中描述。对于遵循本规范的实现，这个branch参数的值必须用magic cookie“z9hG4bK”打头（8.1.1.7节）。

Warning头域用来给应答的状态添加附加说明使用的。Warning头域值是在应答中包含的，并且包括了一个3位的警告代码，主机名，和警告正文。“warn-text”应当是一个自然语言，给个人用户接收应答时候来响应的。

第一个warn－code的数字是”3”表示这是一个SIP规范的警告信息。警告信息300到329是保留用于标志在会话描述中的保留字错误的，330到339是会话描述中基本网络服务相关警告，370到379是关于会话描述重的QoS参数数量相关的警告，390到399是上边未列除的杂项警告信息。

300 Incompatible network protocol:（不兼容的网络协议）

301 Incompatible network address formats(不兼容的网络地址格式)

302 Incompatible transport portocol(不兼容的通讯协议)

303：Incompatible bandwidth units（不兼容的带宽单位）

304 Media type not available（媒体类型不存在）

305 Incompatible media format（媒体格式不兼容）

306 Attribute not understood（媒体属性不支持）

307 Session description parameter not understood（会话描述参数不支持）

330 Multicast not available（多点传输不允许）

331 Unicast not available（Unicast不支持）

370 Insufficient bandwidth(带宽不足)

399 Miscellaneous warning（杂项警告）

1xx和2xx消息是HTTP/1.1使用的。

WWW-Authenticate头域包含了认证信息。

**（第21节） 应答代码 Response Codes**

应答码是包含了，并且扩展了HTTP/1.1应答码。并不是所有的HTTP/1.1应答码都适当应用，只有在这里指出的是适当的。

**临时应答1xx**

也就是消息性质的应答，标志了对方服务器正在处理请求，并且还没有决定最后的应答。如果服务器处理请求需要花200ms以上才能产生终结应答的时候，它应当发送一个1xx应答。

注意1xx应答并不是可靠传输的。他们不会导致客户端传送一个ACK应答。临时性质的（1xx）应答可以包含消息体，包含会话描述。

100 Trying 这个应答表示下一个节点的服务器已经接收到了这个请求并且还没有执行这个请求的特定动作（比如，正在打开数据库的时候）。这个应答，就像其他临时应答一样，终止了UAC重新传送INVITE请求。100(Trying)应答和其他临时应答不同的是，在这里，它永远不会被有状态proxy转发到上行流中。

180 Ringing UA收到INVITE请求并且试图提示给用户。这个应答应当初始化一个本地回铃。

181 Call is Being Forwarded(呼叫被转发) 服务器可以用这个应答代码来表示呼叫正在转发到另一个目的地集合。

182 Queued 当呼叫的对方暂时不能接收呼叫的时候，并且服务器决定将呼叫排队等候，而不是拒绝呼叫的时候，那么就应当发出这个应答。当被叫方一旦恢复接收呼叫，他会返回合适的终结应答。

183（Session Progress）会话进度 应答用于提示建立对话的进度信息。

**成功信息2xx**

这个应答表示请求是成功的。

200 OK 请求已经处理成功。这个信息取决于不同方法的请求的应答。

**转发请求3XX**

3xx系列的应答是用于提示用户的新位置信息的，或者为了满足呼叫而转发的额外服务地点。

300 Multiple Choices 请求的地址有多个选择，每个选择都有自己的地址，用户或者（UA）可以选择合适的通讯终端，并且转发这个请求到这个地址。 应答可以包含一个具有每一个地点的在Accept请求头域中允许的资源特性，这样用户或者UA可以选择一个最合适的地址来转发请求。

UA可以使用Contact头域来自动转发或者要求用户确认转发。

如果被叫方可以在多个地址被找到，并且服务器不能或者不愿意转发请求的时候，可以使用这个应答来给呼叫方。

301 Moved Permently 当不能在Request-URI指定的地址找到用户的时候，请求的客户端应当使用Contact头域(20.10)所指出的新的地址重新尝试。请求者应当用这个新的值来更新本地的目录，地址本，和用户地址cache，并且在后续请求中，发送到这个/这些列出的地址。

302 Moved Temporarily 请求方应当把请求重新发到这个Contact头域所指出的新地址(20.10)。新请求的Request-URI应当用这个应答的Contact头域所指出的值。

305 Use Proxy 请求的资源必须通过Contact头域中指出的proxy来访问。Contact头域指定了一个proxy的URI。接收到这个应答的对象应当通过这个proxy重新发送这个单个请求。305（UseProxy）必须是UAS产生的。

380 Alternative Service 呼叫不成功，但是可以尝试另外的服务。另外的服务在应答的消息体中定义。

**请求失败4xx**

4xx应答定义了特定服务器响应的请求失败的情况。客户端不应当在不更改请求的情况下重新尝试同一个请求。（例如，增加合适的认证信息）。

Client-Error = "400" ; Bad Request

/ "401" ; Unauthorized

/ "402" ; Payment Required

/ "403" ; Forbidden

/ "404" ; Not Found

/ "405" ; Method Not Allowed

/ "406" ; Not Acceptable

/ "407" ; Proxy Authentication Required

/ "408" ; Request Timeout

/ "410" ; Gone

/ "413" ; Request Entity Too Large

/ "414" ; Request-URI Too Large

/ "415" ; Unsupported Media Type

/ "416" ; Unsupported URI Scheme

/ "420" ; Bad Extension

/ "421" ; Extension Required

/ "423" ; Interval Too Brief

/ "480" ; Temporarily not available

/ "481" ; Call Leg/Transaction Does Not Exist

/ "482" ; Loop Detected

/ "483" ; Too Many Hops

/ "484" ; Address Incomplete

/ "485" ; Ambiguous

/ "486" ; Busy Here

/ "487" ; Request Terminated

/ "488" ; Not Acceptable Here

/ "491" ; Request Pending

/ "493" ; Undecipherable

**Server Failure 5xx**

5xx应答是当服务器本身故障的时候给出的失败应答。

Server-Error = "500" ; Internal Server Error

/ "501" ; Not Implemented

/ "502" ; Bad Gateway

/ "503" ; Service Unavailable

/ "504" ; Server Time-out

/ "505" ; SIP Version not supported

/ "513" ; Message Too Large

**Global Failures 6xx**

6xx应答意味这服务器给特定用户有一个最终的信息，并不只是在Request-URI的特定实例有最终信息。

Global-Failure = "600" ; Busy Everywhere

/ "603" ; Decline

/ "604" ; Does not exist anywhere

/ "606" ; Not Acceptable

**（第22节） 使用HTTP认证 Usage of HTTP Authentication**

SIP为认证系统提供了一个无状态的，试错机制，这个认证机制式基于HTTP的认证机制的。任何时候proxy服务器或者UA接收到一个请求（22.1节例外），它尝试检查请求发起者提供的身份确认。当发起方身份确认了，请求的接受方应当确认这个用户是否式通过认证的。在本文档中，没有建议或者讨论认证系统。

本节描述的“Digest”认证机制，只提供了消息认证和复查保护，没有提供消息完整性或者机密性的保证。上述的保护级别和基于这些Digest提供的保护，可以防止SIP攻击者改变SIP请求和应答。

注意由于这个脆弱的安全性，我们不赞成“Basic”（基本的）认证方法。服务器必须不能接收验证方法式“Basic”类型的信任书，并且服务器必须拒绝“Basic”。

SIP认证的框架和HTTP非常接近(RFC2617[17])。特别式，auth-scheme的BNF范式，auth-param,challenge,realm,realm-value,以及信任书都是一样的（虽然对“Basic”认证方案是不允许的）。在SIP，UAS使用401(Unauthorized)应答来拒绝UAC的身份（或者讲是考验UAC的身份，如果不通过，就是401）。另外，注册服务器，转发服务器可以使用401（Unauthorized）来应答身份认证，但是proxy必须不能用401，只能用407（Proxy Authentication Required）应答。

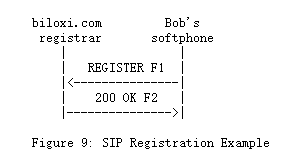
**（第23节） S/MIME 多用途网际邮件扩充协议**

SIP消息可以加载一个MIME 消息体，并且MIME标准包括了MIME内容的保密机制，确保完整性和机密性。实现中应当注意，不管怎样，也会有很少的网络节点（不是典型的proxy服务器），会依赖于查看修改SIP消息（特别是SDP），采用加密的MIME可以防止这类网络节点操作。 这个实现特别对某些类型防火墙有效。

头域可以被proxy服务器合法修改的是：Request-URI， Via， Record-Route，Route，Max-Forwards和Proxy-Authorization。如果这些头域不完全相等，实现中不应当认为是一个安全错误。

**（第24节）Examples**

**Registration**



F1 REGISTER Bob -> Registrar

REGISTER sip:registrar.biloxi.com SIP/2.0

Via: SIP/2.0/UDP bobspc.biloxi.com:5060;branch=z9hG4bKnashds7

Max-Forwards: 70

To: Bob <sip:bob@biloxi.com>

From: Bob <sip:bob@biloxi.com>;tag=456248

Call-ID: 843817637684230@998sdasdh09

CSeq: 1826 REGISTER

Contact: <sip:bob@192.0.2.4>

Expires: 7200

Content-Length: 0

F2 200 OK Registrar -> Bob

SIP/2.0 200 OK

Via: SIP/2.0/UDP bobspc.biloxi.com:5060;branch=z9hG4bKnashds7

;received=192.0.2.4

To: Bob <sip:bob@biloxi.com>;tag=2493k59kd

From: Bob <sip:bob@biloxi.com>;tag=456248

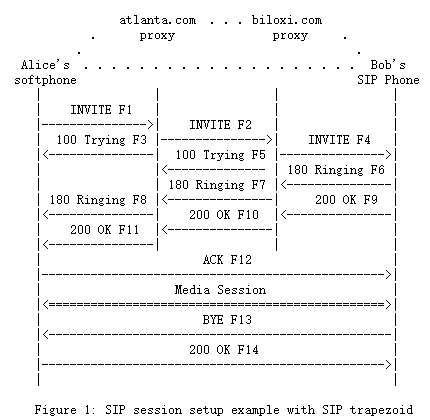
Call-ID: 843817637684230@998sdasdh09

CSeq: 1826 REGISTER

Contact: <sip:bob@192.0.2.4>

Expires: 7200

Content-Length: 0



F1 INVITE Alice -> atlanta.com proxy

INVITE sip:bob@biloxi.com SIP/2.0

Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8

Max-Forwards: 70

To: Bob <sip:bob@biloxi.com>

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 314159 INVITE

Contact: <sip:alice@pc33.atlanta.com>

Content-Type: application/sdp

Content-Length: 142

(Alice's SDP not shown)

F2 100 Trying atlanta.com proxy -> Alice

SIP/2.0 100 Trying

Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8

;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 314159 INVITE

Content-Length: 0

F3 INVITE atlanta.com proxy -> biloxi.com proxy

INVITE sip:bob@biloxi.com SIP/2.0

Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1

Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8

;received=192.0.2.1

Max-Forwards: 69

To: Bob <sip:bob@biloxi.com>

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 314159 INVITE

Contact: <sip:alice@pc33.atlanta.com>

Content-Type: application/sdp

Content-Length: 142

(Alice's SDP not shown)

F4 100 Trying biloxi.com proxy -> atlanta.com proxy

SIP/2.0 100 Trying

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=z9hG4bKnashds8

;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 314159 INVITE

Content-Length: 0

F5 INVITE biloxi.com proxy -> Bob

INVITE sip:bob@192.0.2.4 SIP/2.0

Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1

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=z9hG4bKnashds8

;received=192.0.2.1

Max-Forwards: 68

To: Bob <sip:bob@biloxi.com>

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 314159 INVITE

Contact: <sip:alice@pc33.atlanta.com>

Content-Type: application/sdp

Content-Length: 142

(Alice's SDP not shown)

F6 180 Ringing Bob -> biloxi.com proxy

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1

;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=z9hG4bKnashds8

;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>;tag=a6c85cf

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

Contact: <sip:bob@192.0.2.4>

CSeq: 314159 INVITE

Content-Length: 0

F7 180 Ringing biloxi.com proxy -> atlanta.com proxy

SIP/2.0 180 Ringing

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=z9hG4bKnashds8

;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>;tag=a6c85cf

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

Contact: <sip:bob@192.0.2.4>

CSeq: 314159 INVITE

Content-Length: 0

F8 180 Ringing atlanta.com proxy -> Alice

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8

;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>;tag=a6c85cf

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

Contact: <sip:bob@192.0.2.4>

CSeq: 314159 INVITE

Content-Length: 0

F9 200 OK Bob -> biloxi.com proxy

SIP/2.0 200 OK

Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1

;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=z9hG4bKnashds8

;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>;tag=a6c85cf

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 314159 INVITE

Contact: <sip:bob@192.0.2.4>

Content-Type: application/sdp

Content-Length: 131

(Bob's SDP not shown)

F10 200 OK biloxi.com proxy -> atlanta.com proxy

SIP/2.0 200 OK

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=z9hG4bKnashds8

;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>;tag=a6c85cf

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 314159 INVITE

Contact: <sip:bob@192.0.2.4>

Content-Type: application/sdp

Content-Length: 131

(Bob's SDP not shown)

F11 200 OK atlanta.com proxy -> Alice

SIP/2.0 200 OK

Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8

;received=192.0.2.1

To: Bob <sip:bob@biloxi.com>;tag=a6c85cf

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 314159 INVITE

Contact: <sip:bob@192.0.2.4>

Content-Type: application/sdp

Content-Length: 131

(Bob's SDP not shown)

F12 ACK Alice -> Bob

ACK sip:bob@192.0.2.4 SIP/2.0

Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds9

Max-Forwards: 70

To: Bob <sip:bob@biloxi.com>;tag=a6c85cf

From: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 314159 ACK

Content-Length: 0

The media session between Alice and Bob is now established.

Bob hangs up first. Note that Bob's SIP phone maintains its own CSeq

numbering space, which, in this example, begins with 231. Since Bob

is making the request, the To and From URIs and tags have been

swapped.

F13 BYE Bob -> Alice

BYE sip:alice@pc33.atlanta.com SIP/2.0

Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10

Max-Forwards: 70

From: Bob <sip:bob@biloxi.com>;tag=a6c85cf

To: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 231 BYE

Content-Length: 0

F14 200 OK Alice -> Bob

SIP/2.0 200 OK

Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10

From: Bob <sip:bob@biloxi.com>;tag=a6c85cf

To: Alice <sip:alice@atlanta.com>;tag=1928301774

Call-ID: a84b4c76e66710

CSeq: 231 BYE

Content-Length: 0

**（第25节） SIP协议的BNF范式 Augmented BNF for the SIP Protocol**

本协议中定义的机制都用文本和Backus-Naur Form(BNF)范式定义。

**（第26节） 安全考虑：威胁模式和安全应用建议**

SIP不是一个容易进行安全保护的协议。它使用的中间媒体，以及它的多面信任关系，它希望的节点之间交互基于互不信任的关系，它的用户到用户的操作使得安全保证非常重要。

注册服务

模仿一个服务器

修改消息包体

破坏会话

拒绝服务和扩展

从上边讲述的威胁来看，我们得到了SIP协议所需要的基本安全服务，他们是：保护消息的隐私性和完整性，保护重现（replay）攻击或者消息欺骗，提供会话参与者的身份认证和隐私保护，保护拒绝服务攻击。SIP消息中的包体分别要求机密性，完整性和身份认证。 比为SIP重新定义新的安全机制更好的是，SIP可以重用已经存在的HTTP或者SMTP的安全机制。

**（第27节）IANA 认证**

**（第28节）同RFC 2543的改变**