

# 第十二讲

# SIP协议

自学讲义

# 本讲内容

- ◆ SIP协议综述
- ◆ SIP协议网络体系
- ◆ SIP消息
- ◆ SIP操作
- ◆ SIP呼叫处理
- ◆ SIP会话呼叫编程
- ◆ SIP协议的安全性

# What is SIP

- ◆ The Session Initiation Protocol (SIP) is an application-layer control (signaling) protocol for creating, modifying and terminating sessions with one or more participants. These sessions include Internet multimedia conferences, Internet telephone calls and multimedia distribution.
- ◆ Members in a session can communicate via multicast or via a mesh of unicast relations, or a combination of these.

# SIP是信令协议

- ◆ SIP作为一个应用层的多媒体会话信令协议，可以被用来发起一个会话进程、在会话中邀请其他参加者加入会议。
- ◆ SIP协议主要用于语音与数据相结合的业务、多媒体业务的呼叫建立与释放。
- ◆ SIP协议可以与其他用于建立呼叫的信令系统或协议结合使用，它在设计上充分考虑了对其他协议的可扩展性。

# SIP协议目的

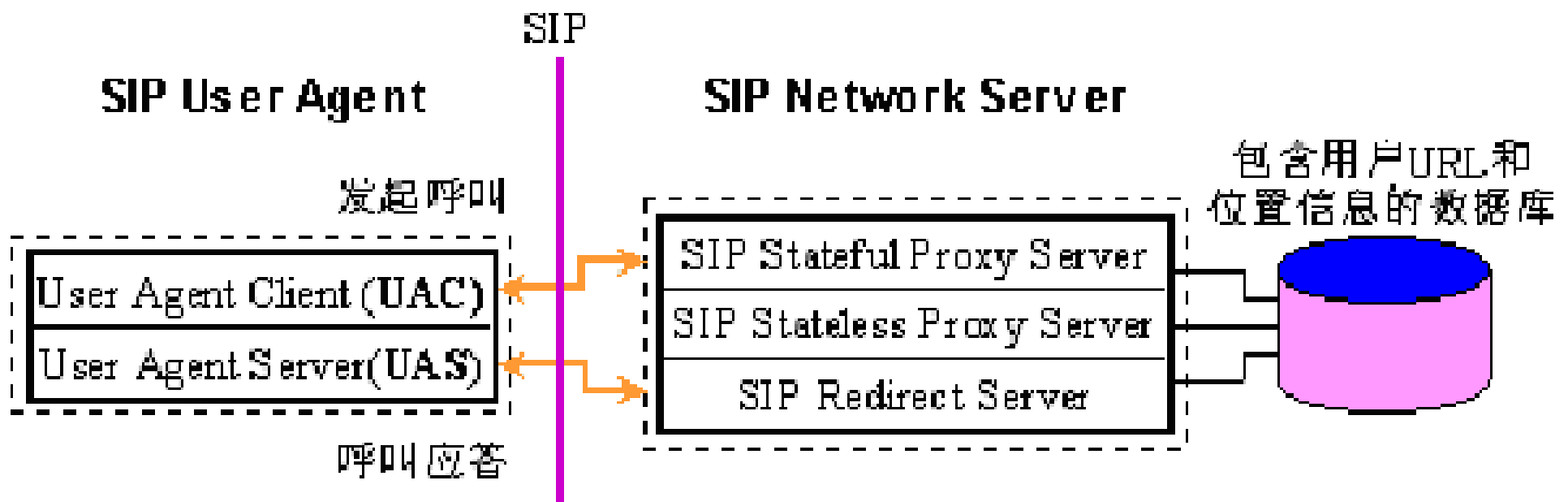
SIP协议是想借鉴Web的成功经验，它通过使用SIP终端将网络设备的复杂性推向网络的边沿，同时SIP可以充分利用已定义的头域，对其进行简单必要的扩充就能很方便地支持各项新业务和智能业务，有利于与Internet的各项应用集成开发VoIP的增值业务。

# SIP协议原则

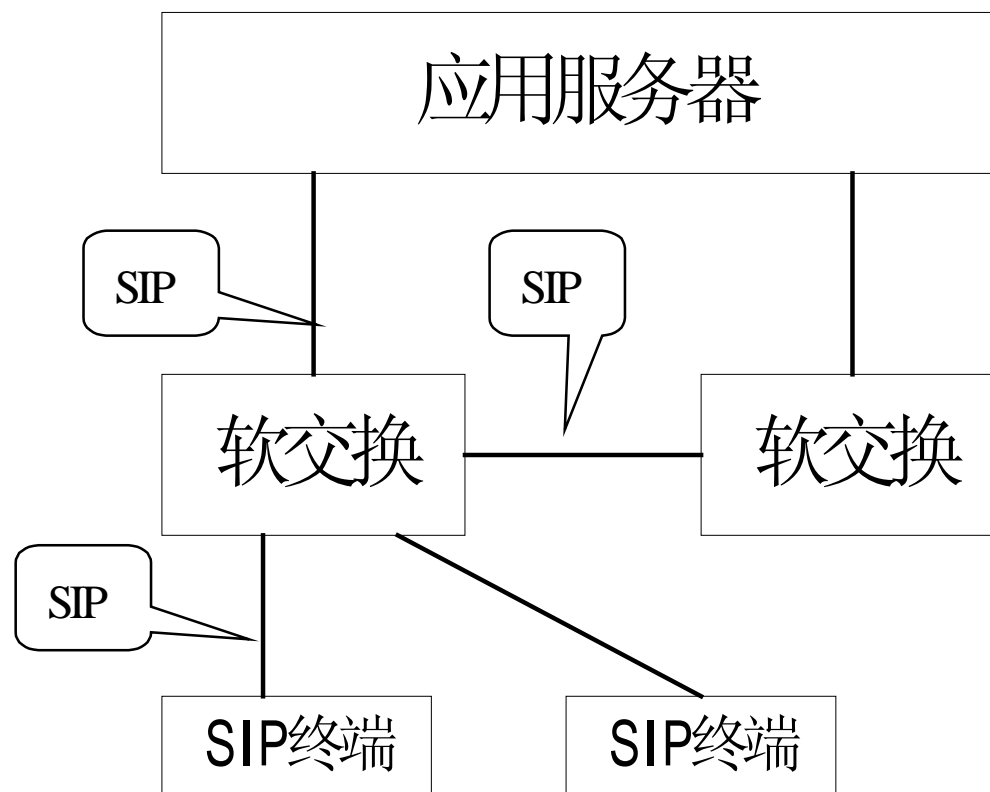
- ◆ SIP协议遵循因特网一贯坚持的简练、开放、兼容和可扩展等原则，并充分注意到因特网开放而复杂的网络环境下的安全问题。
- ◆ SIP协议充分考虑了对传统公共电话网的各种业务，包括IN业务和ISDN业务的支持。

# 客户-服务器方式

- ◆ SIP协议采用基于文格式本的客户-服务器方式，以文本的形式表示消息的语法、语义和编码。
- ◆ SIP协议主要用于SIP终端和软交换之间、软交换和软交换之间以及软交换与应用服务器之间。



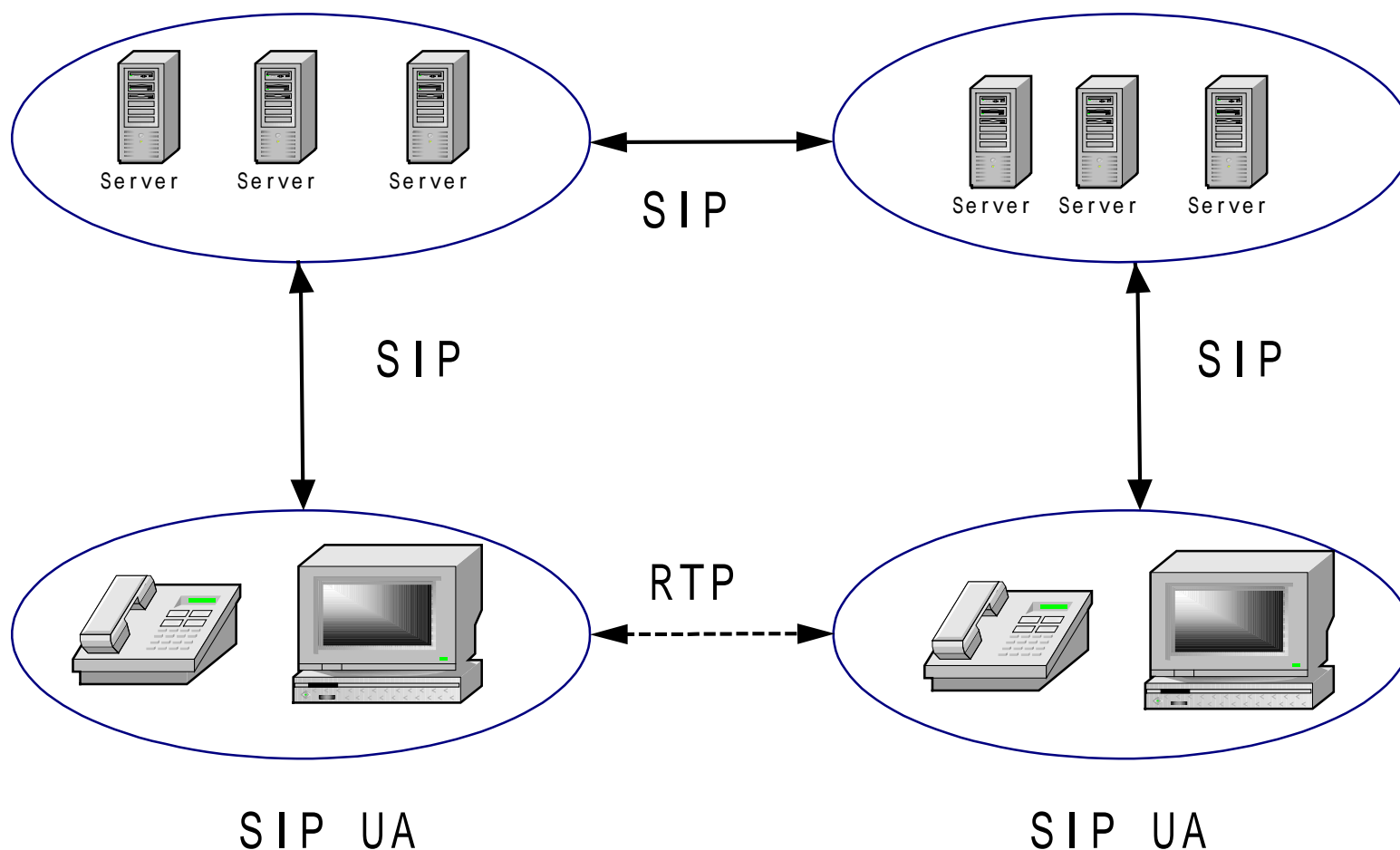
# Where is SIP





# SIP体系结构

SIP注册、代理、重定向服务器



# SIP的基本功能信令功能

- ◆ 用户定位(User location)，确定参加通信的终端用户的位置。
- ◆ 用户能力(User capability)，确定通信采用的媒体类型和参数。
- ◆ 用户可用性(User availability)，确定被叫是否愿意加入通信过程。
- ◆ 呼叫建立(Call setup)，包括向被叫“振铃”，确定主叫和被叫的连接参数。
- ◆ 呼叫处理(Call handling)，包括呼叫重定向、呼叫转移、终止呼叫等等。

# SIP网络组件

SIP网络包含两类组件：

用户代理(User Agent)

网络服务器(Network Server)

# 用户代理(User Agent)

- ◆ 用户代理又分为用户代理客户端(UAC)和服务端(UAS):

UAC负责发起SIP呼叫请求

UAS负责对呼叫请求作出相应

- ◆ 目前设备厂商开发的SIP智能终端或软终端通常包含UAC和UAS两种功能。

# 网络服务器(Network Server)

- ◆ SIP网络服务器主要为用户代理提供注册，认证，鉴权，路由等服务，分为代理服务器（Proxy），重定向服务器（Redirect Server）和注册服务器（Register）。

# 代理服务器（Proxy）

- ◆ Proxy提供路由功能，代理其他客户机发起的请求，请求由本地服务器响应或可能被翻译之后再传送给其他服务器。
- ◆ 代理服务器在转发请求之前需要对原请求消息进行解释，而且必要的话则还必须重写原请求消息。
- ◆ Proxy分为有状态stateful代理和无状态stateless代理两种。

# Stateful Proxy

- ◆ 有状态代理（Stateful Proxy）记录转发呼叫的状态信息。
- ◆ Stateful Proxy通常位于SIP网络的边缘。为SIP网络的运营管理提供相关的信息。

# Stateless Proxy

- ◆ 无状态代理（Stateless Proxy）不记录转发呼叫的状态信息。
- ◆ 无状态代理通常位于SIP网络的核心，核心Proxy需要处理大量的呼叫，不保留呼叫状态可以大大提高系统的处理能力。



# 重定向服务器（Redirect Server）

- ◆ 重定向服务器是一个接收SIP请求、把该地址映射成零个或多个新地址并把这些地址返回给请求客户。
- ◆ 不同于代理服务器，重定向服务器不发起它自己的SIP请求。
- ◆ 不同于用户代理服务器，重定向服务器不接受呼叫。

# 注册服务器（Register）

- ◆ 注册服务器接受终端的Register请求。
- ◆ 用户终端在启动后都需要进行注册，记录
- ◆ 一个注册服务器通常和一个代理或重定向服务器位于同一物理实体中。并可提供定位服务（Location Services）。

# 定位服务器（Location Server）

- ◆定位服务器（Location Server）提供定位服务，为SIP重定向和代理服务器获得被叫方的可能位置信息（被呼叫用户的地址）。
- ◆Location Server可以和SIP网络服务器结合在一起，但定位服务器并不属于SIP服务器范畴。

# SIP的应用

- ◆ SIP用来发起一个会话进程、在会话中邀请其他参加者加入会议。
- ◆ SIP协议支持别名映射、重定向服务、ISDN和IN业务。它支持个人移动（personal mobility），即终端用户能够在任何地方、任何时间请求和获得已订购的任何电信业务。
- ◆ SIP协议可以通过MCU、单播联网方式、或组播方式创建多方会话，支持PSTN和因特网电话之间的网关功能。

# SIP服务器与其他后台应用配合

- ◆ SIP服务器需通过LDAP与定位服务器通信为终端提供号码查询服务
- ◆ 通过RADIUS协议与RADIUS服务器通信为终端通过认证，鉴权
- ◆ 通过XML实现对业务的计费

# SIP与会议控制

- ◆ SIP协议不提供发言控制（floor control）、投票等会议控制功能，也不规定如何管理一个会议。但是SIP协议可被用来引发这些会议控制协议。
- ◆ SIP协议本身不具备资源预留功能，但可以向被邀请者们传达这方面的信息。

# SIP协议与多媒体通信系统

- ◆ SIP协议可以与其他协议一起构建因特网多媒体通信系统。这些协议包括RSVP、RTP/RTCP、SDP（会话描述协议）、SAP（会话通告协议）、RTSP（实时流协议）、SCCP（简单会议控制协议）等。
- ◆ SIP协议所规范的操作和相应的功能是协议独立的。
- ◆ SIP独立于下面的传输层协议，可以灵活方便地扩展其他附加功能

# SIP消息（message）

- ◆ SIP消息是SIP客户终端和服务端之间通信的基本信息单元。
- ◆ SIP消息基于文本，采用UTF-8编码（RFC 2279）中的ISO 10646字符集。
- ◆ SIP协议借鉴了HTTP协议（RFC 2068）的设计思想，有很多消息格式与之相同。
- ◆ SIP协议支持UDP传输协议。



# SIP消息分类

◆SIP消息分为两类：

UAC到UAS的请求（Request）

UAS到UAC的响应（Response）

SIP-message = Request | Response

# SIP消息格式

SIP消息由一个起始行（Start-line）、一个或多个字段(header fields)组成的消息头、一个标志消息头结束的空行(CRLF)以及作为可选项的消息体(Message body)组成，其中描述的头称为实体头(Entity header)。

```
generic-message = start-line  
                  *message-header  
                  CRLF  
                  [ message-body ]
```

# 起始行（Start-line）

◆ 起始行分请求行（Request-Line）和状态行（Status-Line）两种。

n 请求行：请求消息的起始行。

n 状态行：响应消息的起始行。

start-line = Request-Line | Status-Line

# 消息头

◆ 消息头分通用头（general-header）、请求头（request-header）、响应头（response-header）和实体头（entity-header）四大类36种。

message-header = ( general-header  
| request-header  
| response-header  
| entity-header )

# 四大类消息头说明

- ◆ general-header为描述消息基本属性的通用头域，可用于请求消息和应答消息；
- ◆ request-header为请求头域，只可用于请求消息，它被用来传递有关应答的附加信息，对请求进行补充说明；
- ◆ response-header为应答头域，只可用于应答消息，它被用来传递有关应答的附加信息，对应答进行补充说明。
- ◆ entity-header是消息体头域，用于描述消息体内容的长度、格式和编码类型等属性，可用于请求消息或应答消息。

# general-header类消息头

◆ Call-ID, From, To, Via, Contact, CSeq, Encryption, Expires, Record-Route, Timestamp, Date, Accept, Accept-Encoding, Accept-Language

◆ 详细内容请见SIP协议（RFC 2543）第6章。

# entity-header类消息头

◆Content-Encoding, Content-Length, Content-Type 。

◆详细内容请见SIP协议（RFC 2543）第6章。

# request-header类消息头

◆ Subject, User-Agent, Organization, Contact, Authorization, Proxy-Authorization, Proxy-Require, Response-Key, Require, Priority, Hide, Route, Max-Forwards。

◆ 详细内容请见SIP协议（RFC 2543）第6章。



# response-header类消息头

◆ Proxy-Authenticate, WWW-Authenticate, Retry-After, Server, Warning, Allow, Unsupported。

◆ 详细内容请见SIP协议（RFC 2543）第6章。

# 消息头格式

- ◆消息头格式遵循RFC 822 Internet文本消息格式标准中的头域格式规范。每个消息头都是一个“句子”，以CRLF行结束符表示一个头域的结束。
- ◆它们都由字段名（field-name）和域值（field-value）两部分组成，中间以“:”相隔。一般对域值的规定和解释与具体的各个消息头名有关。

# 消息头格式举例

- ◆ Accept: application/sdp; level=1,application/X-private,text/html
- ◆ Call-ID: 1234567123@bupt.edu.cn
- ◆ Content-Length: 2345
- ◆ Hide: route
- ◆ Contact:  
zhang@mail.tinghua.edu.cn;tag=123;q=0.7,maito:lee@bupt.edu.cn
- ◆ To: tel:010-6228-1234
- ◆ From:  
62281234@IPPhoneGateway.BTA.com.cn;user=phone
- ◆ Server: 北京邮电大学SIP会话代理服务器。

# SIP请求消息

SIP请求消息的定义格式如下：

Request = Request-Line

\*( general-header

| request-header

| entity-header )

CRLF

[ message-body ]

# 请求行（ Request-Line ）

请求行（ Request-Line ）以一个方法标识词Method开始，接着是请求的目的发送地址Request-URI，SIP协议的版本号，最后是行结束符CRLF：

Request-Line = Method SP Request-URI SP  
SIP-Version CRLF

# SIP地址

- ◆ SIP请求消息的start-line中Request-URI部分是一个SIP URL，它表示这个请求消息发送的用户或服务的当前目的地址。
- ◆ 当请求经由代理服务器转发时，SIP URL可能被代理服务器改写，所以我们可以认为Request-URI类似于网络层的“下一站地址”。
- ◆ 真正的被叫地址包含在To头域中。
- ◆ Via地址列表记录了一个请求消息的路由（的信息，Via列表中的第一项是发起请求的主叫地址。
- ◆ From头域包含的是逻辑主叫的地址。

# SIP方法（Methods）

◆ Method是SIP请求消息的消息头中的一个重要字段（Field）。

◆ SIP定义了7种方法：

Method = "INVITE"  
| "ACK"  
| "OPTIONS"  
| "BYE"  
| "CANCEL"  
| "REGISTER"

# INVITE

- ◆该方法用于邀请用户或服务参加一个会话。
- ◆在该请求的消息体中可对被叫方被邀请参加的会话加以描述，如主叫方能接受的媒体类型，发出的媒体类型和一些参数；对该请求的成功响应必须在响应的消息体中说明被叫方愿意接受哪种媒体，或者说明被叫方发出的媒体。
- ◆服务器可以自动以200（OK）响应来响应会话邀请。



# ACK

- ◆ ACK方法用于客户机向服务器证实它已经收到了对INVITE请求的最终响应。
- ◆ ACK只和INVITE一起使用。
- ◆ 对2XX最终响应的证实由客户机用户代理发出，对其他最终响应的证实由收到响应的第一个代理或第一个客户机用户代理。
- ◆ ACK请求的TO，FROM，CALL-ID，CSEQ字段的值由对应的INVITE请求的相应字段的值复制而来。

# OPTIONS

- ◆ OPTIONS方法用于向服务器查询其能力。如服务器认为它能与用户联系，则可用一个能力集响应该请求，对于代理和重定向服务器只要转发此请求，不用显示其能力。
- ◆ 该请求中FROM，TO分别包含主被叫的地址信息，对该请求响应中的FROM，TO，CALL-ID字段的值由该请求中相应的字段值复制而来。

# BYE

- ◆ 用户代理客户机用该请求向用户代理服务器表明它想释放呼叫。
- ◆ 该请求可以像INVITE请求那样被转发，可由主叫方发出也可由被叫方发出。
- ◆ 呼叫的一方在释放（挂断）呼叫前必须发出此请求，收到该请求的一方必须停止向发出请求的一方发送媒体流。

# CANCEL

- ◆ CANCEL用于取消一个CALL-ID，TO，FROM，CSEQ字段值相同的正在进行的请求，但取消不了已经完成的请求（如果服务器返回一个最终状态响应，则认为请求已经完成）。
- ◆ 该请求中的CALL-ID，TO，CSEQ的数字部分及FROM字段和原请求的对应字段值相同，从而使该请求与它要取消的请求匹配。

# REGISTER

REGISTER方法用于客户机向SIP服务器注册列在TO字段中的地址信息。

# INFO

INFO方法是对SIP协议的扩展，用于传递会话中产生的与会话相关的控制信息，如ISUP和ISDN信令消息，有关此方法的使用还有待标准化。

# SIP应答消息

- ◆当服务器收到一个SIP请求消息并对其分析理解后，服务器根据具体请求要返回一个或多个SIP应答消息。
- ◆SIP应答消息类似于HTTP应答。

# SIP应答消息格式

Response = Status-Line

\*(general-header  
|response-header  
|entity-header)

CRLF

[message-body];



# SIP应答消息的Status-Line

SIP应答消息的Status-Line由SIP-Version开始，接着是一个数字编码的状态码Status-Code，最后是一个与状态码相关的描述性短语Reason-Phrase，然后由一个CRLF行结束符结束Status-Line。其格式定义如下：

Status-Line = SIP-Version SP Status-Code  
SP Reason-Phrase CRLF

# SIP应答消息的状态码（应答码）

- ◆如同请求消息中的方法标识符被用来区分各种不同请求一样，应答消息中的状态码也是被用来区分各种不同SIP应答的。
- ◆状态码是一个3位十进制整数，用来表示服务器对客户机所发请求的理解和执行结果。

# SIP应答消息的六类应答状态编码

- ◆ **1xx: 临时消息**，表示表示请求消息已经收到，后面将继续处理该请求。
- ◆ **2xx: 成功消息**，表示请求已经被成功的理解、接受或执行。
- ◆ **3xx: 重定向消息**，表示为了完成请求还需采取更进一步的动作。
- ◆ **4xx: 客户机错误**，表示该请求含有语法错误或在这个服务器上不能被满足。
- ◆ **5xx: 服务器错误**：表示该服务器不能处理一个明显有效的请求。
- ◆ **6xx: 全局性故障**，表示该请求在任何服务器上都不能被实现。

# 应答码的具体含义

- ◆ 应答码个位和十位（xx）的进一步编码被用来细分各种情况。
- ◆ 对应答码含义的确切理解和对它的处理将与其的具体“上下文”有关。

# SIP操作

本部分内容：

- ◆ SIP寻址和SIP 通用资源定位器
- ◆ 定位SIP服务器
- ◆ SIP交互
- ◆ 登录服务
- ◆ 会话期间改变会话属性

# SIP寻址和SIP 通用资源定位器

- ◆ SIP协议对位于某个主机的用户使用SIP的通用资源定位器（URL）进行标识，并根据该URL进行寻址。
- ◆ SIP的通用资源定位器采用与mailto和telnet等一致的URL格式，即“主机名+用户名”：`user@host`格式。
- ◆ 用户部分（user）是用户名字或电话号码；主机部分（host）可以是DNS域名（RFC 2052）、CNAME或A记录（RFC 1035）、或者IP地址。

# SIP通用资源定位器示例

◆ sip:mjh@metro.isi.edu

◆ sip:watson@bell-telephone.com

◆ sip:root@193.175.132.42

◆ sip:info@ietf.org

◆ sip:62281234@IPPoneGateway.BTA.com.cn;  
user=phone

◆ sip:sales@ruitong.com.cn

# 定位SIP服务器

- ◆ 发送SIP请求首先要定位SIP服务器，其中SIP请求消息中的Request-URI就表示该请求的当前目的地址。
- ◆ 定位SIP服务器根据请求中的目的SIP地址，即Request-URI中的主机部分确定该请求的下一站服务器的IP地址，完成从Request-URI到服务器IP地址的转换。



# SIP交互

- ◆ Request-URI中的host部分被成功解析为某一SIP服务器的IP地址后，客户机就向该服务器发送一个或多个SIP请求，并从该服务器接收一个或多个应答。
- ◆ 一个请求（包括它的重发）与由这个请求触发的所有应答组成一次SIP交互。

或称为事务


# 登录服务

- ◆ 客户通过登录请求使得代理服务器或重定向服务器知道哪个地址可以到达。
- ◆ 客户还可以使用登录请求在服务器上配置呼叫处理属性（这可以用于实现某些高级的智能业务和ISDN业务：比如呼叫筛选等）。

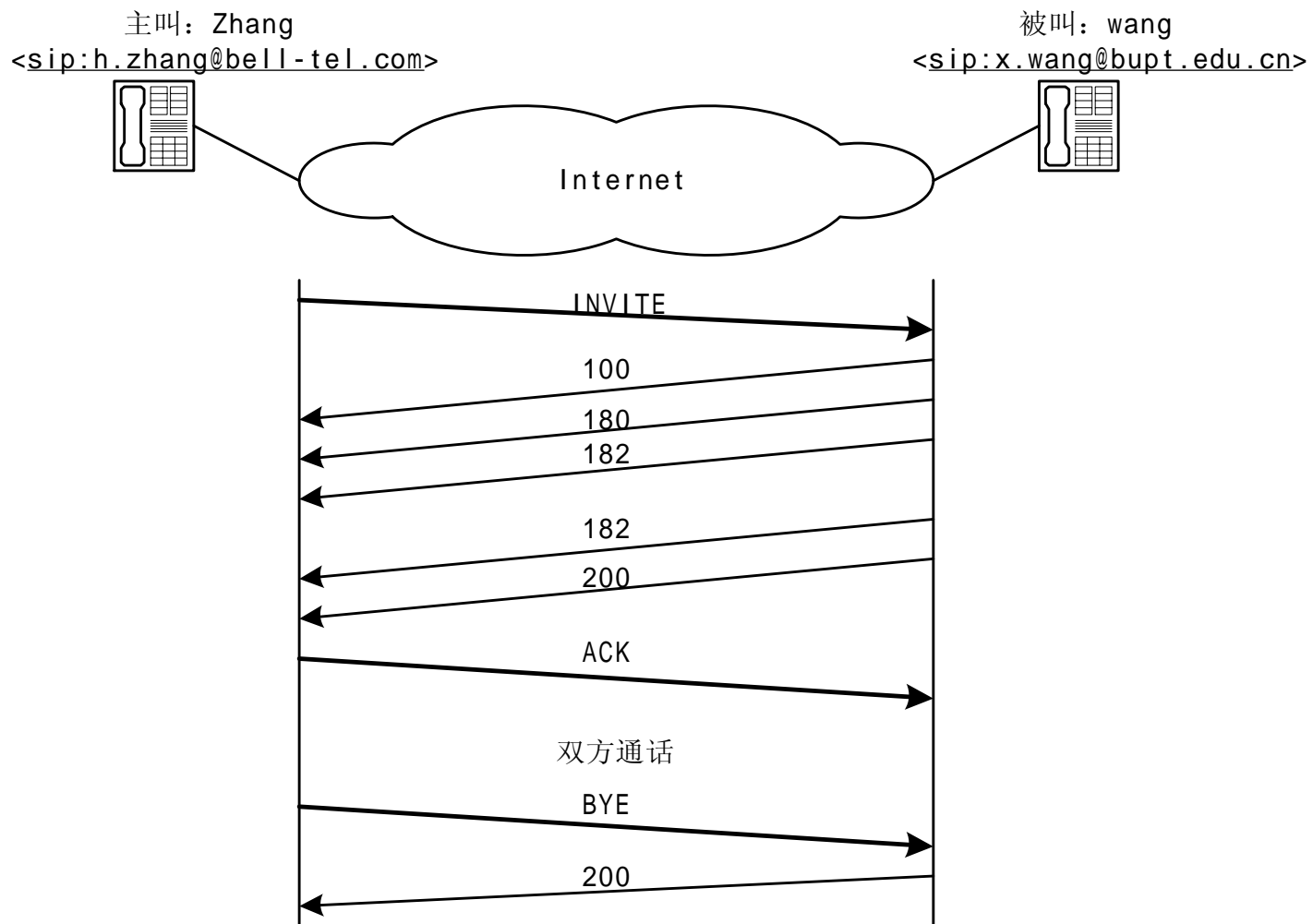
# 会话期间改变会话属性

- ◆ 我们可以通过再发起一个INVITE请求来改变正在进行的会话的参数。（如两个用方会话改为三方会话）
- ◆ 该INVITE请求使用的Call-ID不变，但包含不同的头域值和消息体内容来传递要修改的会话属性，同时使用一个比上次客户机发给服务器的INVITE请求的CSeq值更高的CSeq(command sequence)值。

# 三种SIP呼叫模式

- ◆直接呼叫：由主叫UAC向被叫UAS直接呼叫。
- ◆代理呼叫：由主叫UAC在重定向服务器的辅助下进行重定向呼叫。
- ◆重定向呼叫：由代理服务器代表主叫UAC向被叫UAS发起呼叫。

# 直接呼叫

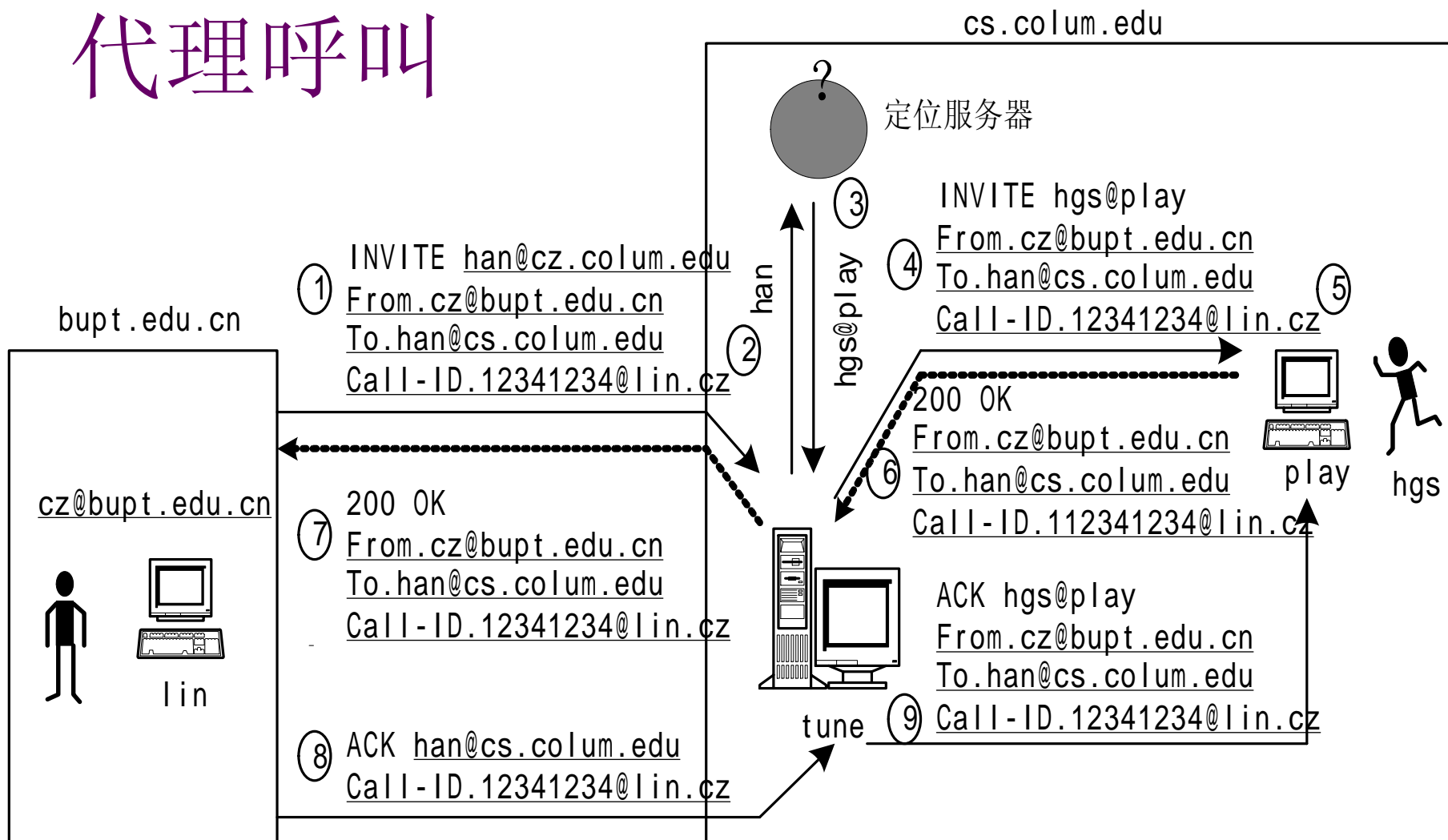


SIP直接呼叫流程示意图

# 直接呼叫流程

- (1)主叫向被叫发出INVITE请求
- (2) 被叫接收到请求后应答
- (3) 主叫收到应答后发送ACK请求
- (4) 主叫或被叫在呼叫建立后发起后续请求

# 代理呼叫



代理服务呼叫示意图

# 代理呼叫流程（一）

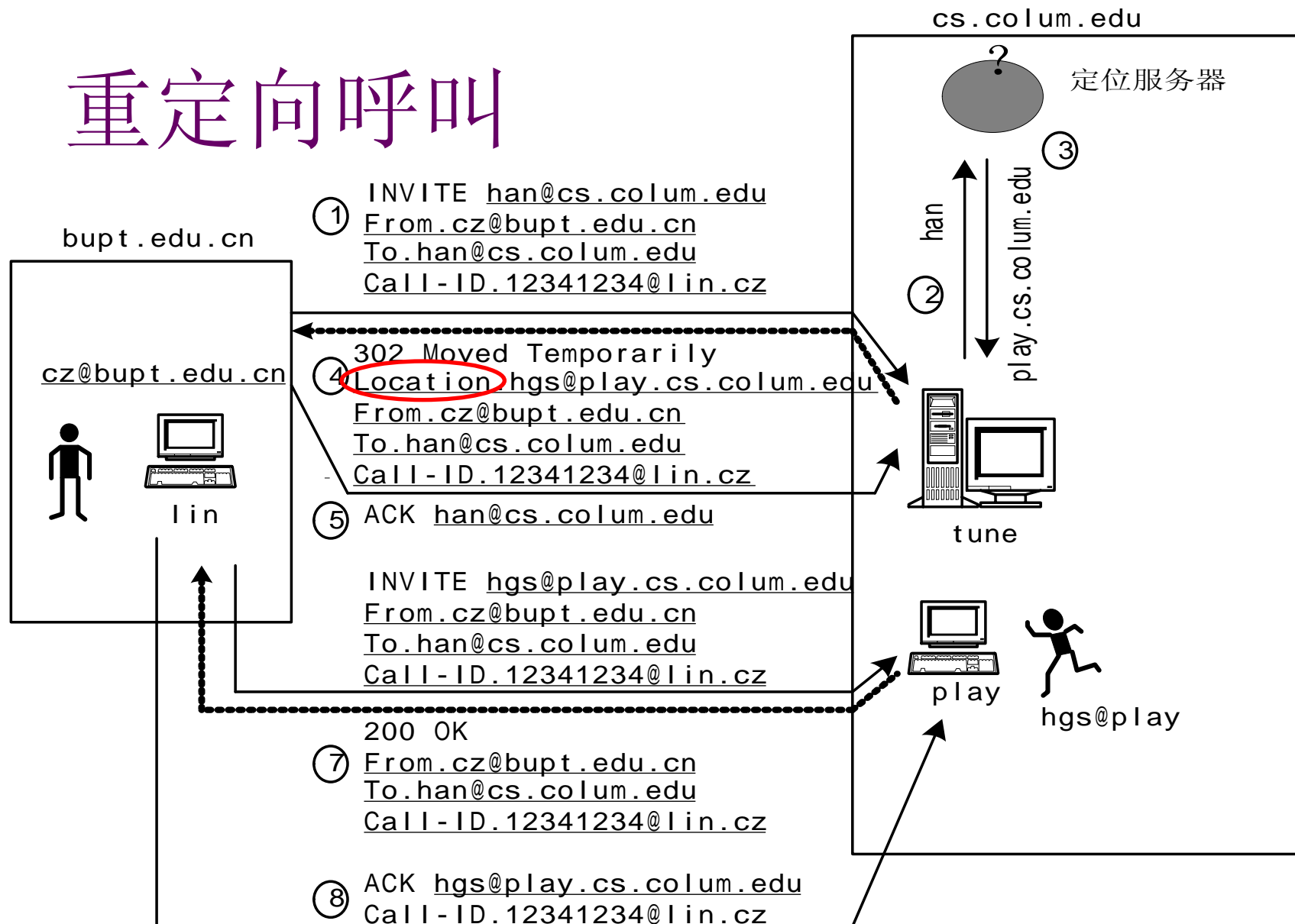
- （1）代理服务器接受INVITE请求。
- （2）使用各种地址方式进行联系定位服务器。
- （3）得到更精确的位置信息。
- （4）然后代理服务器向定位服务器返回的地址发送一个请求。



## 代理呼叫流程（二）

- （5）用户代理服务器提示用户。
- （6）接着向代理服务器返回一个成功的指示。
- （7）代理服务器然后向原主叫返回成功的结果。
- （8，9）主叫收到该消息后，发送ACK请求进行确认，这个ACK确认请求再被转发给被叫。

# 重定向呼叫



重定向呼叫示意图

# 重定向呼叫流程

- (1) 重定向服务器接受INVITE请求。
- (2, 3) 象前面那样进行联系定位服务器（第二、三步）。
- (4) 然后，不是自己直接去联系新找到的地址，而是将该地址返回给主叫。
- (5) 随后通过一个ACK请求得到确认。
- (6) 主叫再向服务器返回的地址发送一个带有相同Call-ID及更高CSeq的新请求。
- (7) 在这个例子中，呼叫获得成功。
- (8) 主叫与被叫以ACK结束握手。

# SIP会话呼叫编程举例

1. 呼叫建立编程
2. 结束呼叫编程

# 呼叫建立说明（一）

- ◆ 位于bell-tel.com主机上的Bell用户（From）将呼叫位于boston.bell-tel.com主机上的Watson用户（To）。
- ◆ Bell向服务器boston.bell-tel.com（Request-URI的host部分）发送INVITE消息，其消息体为一个遵循SDP协议的会话描述（Content-Type），表明他能够接受的媒体类型包括音频编码0（PCM U律编码）、3（GSM语音编码）和4（G.723语音编码）格式
- ◆ 媒体传输采用RTP实时传输协议，要求Watson将媒体发往135.180.144.94地址的3456端口。

## 呼叫建立说明（二）

- ◆ 在这个呼叫例子中，由于INVITE请求消息中没有另外指出（Contact）应答的返回地址，所以被叫服务器向地址kton.bell-tel.com（Via）返回了一系列应答。可能会因进行数据库查询等待处理而引起延迟，所以首先应立刻以100进行应答，接着返回被叫振铃180应答，由于这时被叫端还有两个呼叫在排队，所以应分别先后应答“182, 2 Caller ahead”和“182, 1 Caller ahead”。再后Watson应答200接受Bell的邀请，并且在200应答消息中包含描述被叫Watson所能接受的会话属性；
- ◆ 被叫Watson能够接受0（PCM U律）类型、3（GSM语音编码）类型的媒体编码格式；要求主叫Bell将媒体流发往boston.bell.com；媒体传输采用RTP协议；

# 呼叫建立说明（三）

- ◆ 由于没有特别指出RTP协议的端口则采用默认端口5004接收RTP分组包，端口5005接收RTCP包。
- ◆ 当主叫Bell收到200且成功应答后，由于双方在会话媒体类型等方面已达成一致，所以在Bell接下来发起的ACK回证实请求中不必再包含会话描述。
- ◆ 对客户机发来的回证实请求，服务器不返回应答。并且这个标志SIP呼叫的三次握手结束。在这个例子中，呼叫成功。

# CàS:

- ◆ INVITE sip: watson@boston.bell-tel.com SIP/2.0
- ◆ Via: SIP/2.0/UDP kton.bell-tel.com
- ◆ From: A.Bell<sip:a.g.bell@bell-tel.com>
- ◆ To: T.Watson<sip:watson@bell-tel.com>
- ◆ Call-ID: 1234567234564ade123@kton.bell-tel.com
- ◆ Subject: Mr.Watson, Come here.
- ◆ CSeq: 1 INVITE
- ◆ Content-Type: Application/sdp
- ◆ Content-Length: ...
- ◆ V=0
- ◆ o=bell 53655678 23456789 IN IP4 128.3.4.5
- ◆ c=IN IP4 135.180.144.94
- ◆ m=audio 3456 RTP/AVP 0 3 4



# SàC:

- ◆ SIP/2.0 100 Trying
- ◆ Via: SIP/2.0/UDP kton.bel-tel.com
- ◆ From: A.Bell<sip:a.g.bell@bell-tel.com>
- ◆ To: T.Watson<sip:watson@bell-tel.com>
- ◆ Call-ID: 1234567234564ade123@kton.bel-tel.com
- ◆ CSeq: 1 INVITE
- ◆ Content-Length: 0

# SàC:

- ◆ SIP/2.0 180 Ringing
- ◆ Via: SIP/2.0/UDP kton.bel-tel.com
- ◆ From: A.Bell<sip:a.g.bell@bell-tel.com>
- ◆ To: T.Watson<sip:watson@bell-tel.com>
- ◆ Call-ID: 1234567234564ade123@kton.bell-tel.com
- ◆ CSeq: 1 INVITE
- ◆ Content-Length: 0

# SàC:

- ◆ SIP/2.0 182 Qyeued,2 Caller ahead
- ◆ Via: SIP/2.0/UDP kton.bel-tel.com
- ◆ From: A.Bell<sip:a.g.bell@bell-tel.com>
- ◆ To: T.Watson<sip:watson@bell-tel.com>
- ◆ Call-ID: 1234567234564ade123@kton.bell-tel.com
- ◆ CSeq: 1 INVITE
- ◆ Content-Length: 0

# SàC:

- ◆ SIP/2.0 182 Qyeued,1 Caller ahead
- ◆ Via: SIP/2.0/UDP kton.bel-tel.com
- ◆ From: A.Bell<sip:a.g.bell@bell-tel.com>
- ◆ To: T.Watson<sip:watson@bell-tel.com>
- ◆ Call-ID: 1234567234564ade123@kton.bell-tel.com
- ◆ CSeq: 1 INVITE
- ◆ Content-Length: 0

# SàC:

- ◆ SIP/2.0 200 OK
- ◆ Via: SIP/2.0/UDP kton.bel-tel.com
- ◆ From: A.Bell<sip:a.g.bell@bell-tel.com>
- ◆ To: T.Watson<sip:watson@bell-tel.com>
- ◆ Call-ID: 1234567234564ade123@kton.bel-tel.com
- ◆ CSeq: 1 INVITE
- ◆ Content-Length: ...
- ◆ V=0
- ◆ o=watson 23565678 48596789 IN IP4 192.1.2.3
- ◆ s=I'm on my way
- ◆ c=IN IP4 bosto.bell.com
- ◆ m=audio 5004 RTP/AVP 0 3

# CàS:

- ◆ ACK sip:watson@boston.bell-tel.com  
SIP/2.0
- ◆ Via: SIP/2.0/UDP kton.bel-tel.com
- ◆ From: A.Bell<sip:a.g.bell@bell-tel.com>
- ◆ To: T.Watson<sip:watson@bell-tel.com>
- ◆ Call-ID: 1234567234564ade123@kton.bell-tel.com
- ◆ CSeq: 1 ACK

# 结束呼叫

- ◆ 如果主叫或被叫要结束这次呼叫就向对方发送一个BYE请求消息即可。
- ◆ 其中Call-ID不变，如果是主叫向被叫拆线，From和To不变；如果是被叫向主叫挂机则需要将From和To互换一下，Call-ID仍然不变，请求发送的目的地址则为主叫地址（[a.g.bell@bell-tel.com](mailto:a.g.bell@bell-tel.com)）

下面是主叫Bell向被叫Watson挂机的消息：

# CàS

- ◆ BYE sip:watson@boston.bell-tel.com SIP/2.0
- ◆ Via: SIP/2.0/UDP kton.bel-tel.com
- ◆ From: A.Bell<sip:a.g.bell@bell-tel.com>
- ◆ To: T.Watson<sip:watson@bell-tel.com>
- ◆ Call-ID: 1234567234564ade123@kton.bell-tel.com
- ◆ CSeq: 2 BYE



# CàS:

- ◆ BYE sip: a.g.bell@bell-tel.com SIP/2.0
- ◆ Via: SIP/2.0/UDP kton.bel-tel.com
- ◆ From: T.Watson<sip:watson@bell-tel.com>
- ◆ To: A.Bell<sip:a.g.bell@bell-tel.com>
- ◆ Call-ID: 1234567234564ade123@kton.bell-tel.com
- ◆ CSeq: 1 BYE

# SIP协议的安全性

SIP协议提供了一定的安全手段，可增强SIP呼叫代理的安全性，主要包括加密机制和认证机制。

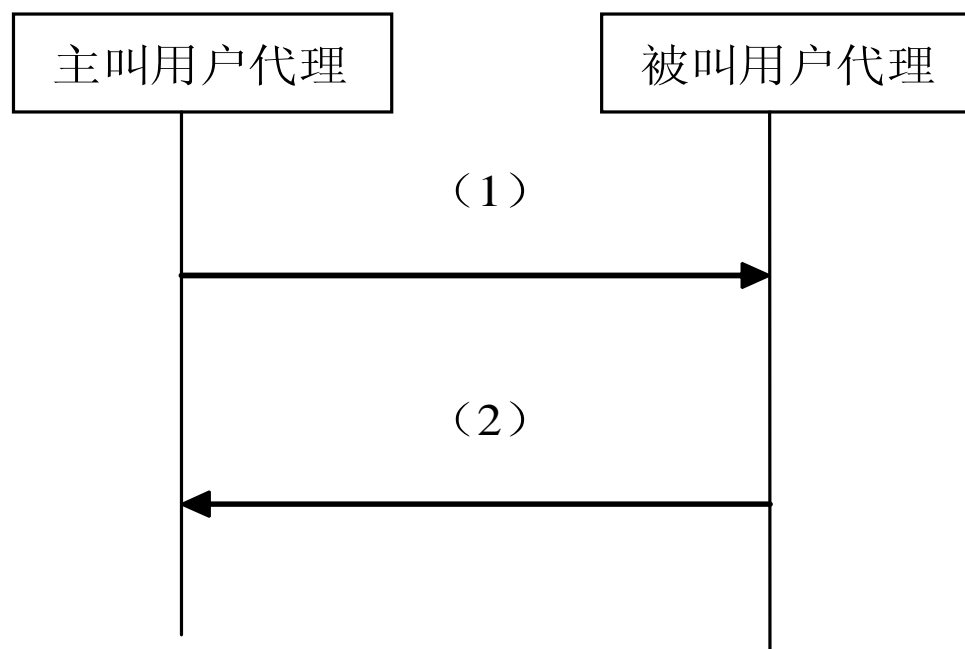
# 加密机制

- ◆ SIP请求和响应消息中含有与通信有关的敏感信息，需采用必要的安全措施保证通信的保密性，这可通过加密机制实现，如果消息经过了加密处理，必须在ENCRYPTION字段表明所采用的加密机制。
- ◆ 加密机制有端到端加密和 跳一跳加密两种。

# 端到端加密

- ◆ 对SIP消息体和某些敏感消息头字段进行端到端加密。
- ◆ 典型方式是用响应方的密钥对请求消息进行加密，用请求方的密钥对响应消息进行加密，所有的实现都必须支持基于PGP的加密机制。

# 端到端加密示意图



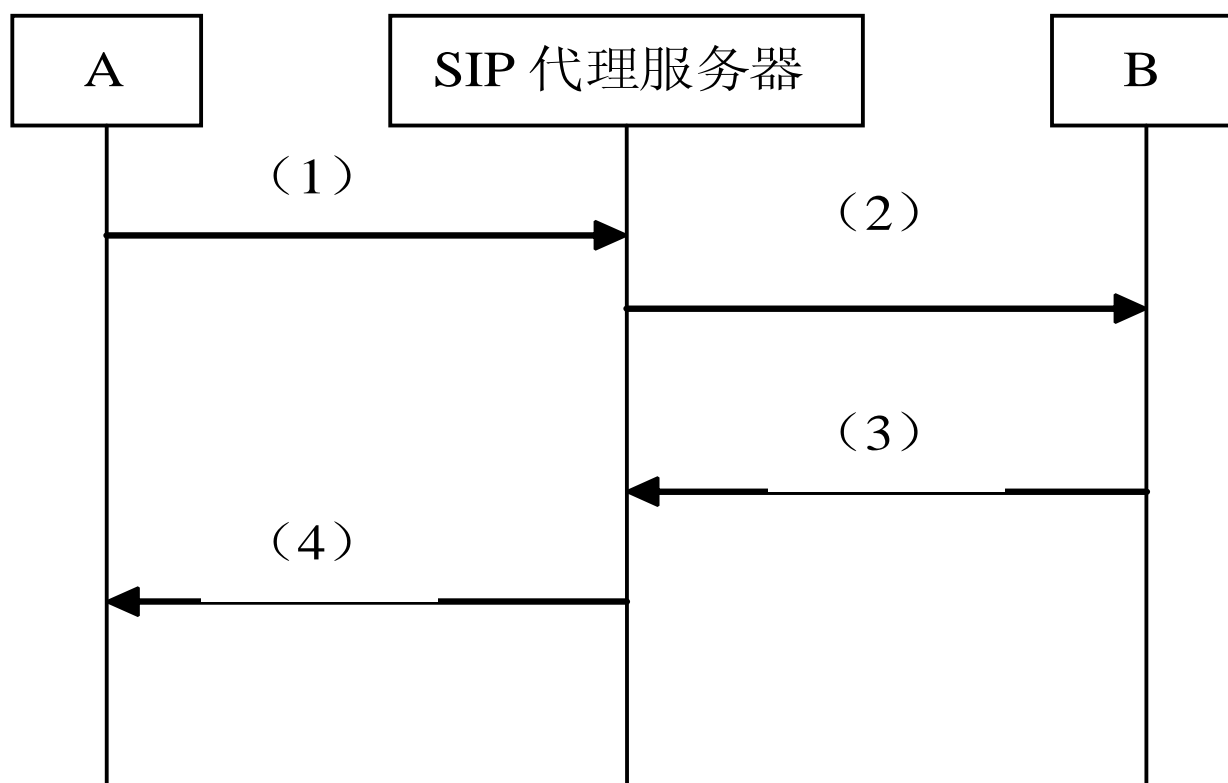
# 密钥的传送流程

- (1) 主叫用户代理在请求消息的RESPONSE-KEY字段中给出主叫方的密钥，被叫用户代理用此密钥对响应消息进行加密。
- (2) 被叫用户代理用200OK响应来响应请求消息，表明收到了主叫方的密钥。

# 跳到跳加密

跳到跳加密主要通过通过对Via字段进行加密实现，用于隐藏请求消息的路由，防止窃听和跟踪。

# 跳到跳加密示意图





# 跳到跳加密流程

- (1) SIP代理服务器收到来自A的请求消息，此消息中含“Hide: hop”，SIP代理服务器知道A希望将“Via: A”向下一服务器B隐藏起来，于是将“Via: A”在代理服务器的缓冲区中存储起来，缓冲区的索引号假定为01。
- (2) 将“Via: 01”发给B，这样B就不知道此请求经过了A。
- (3) 从B向代理服务器返回响应。
- (4) SIP代理服务器收到响应后，从缓冲区01中取出值“Via: A”，然后将响应发给A。

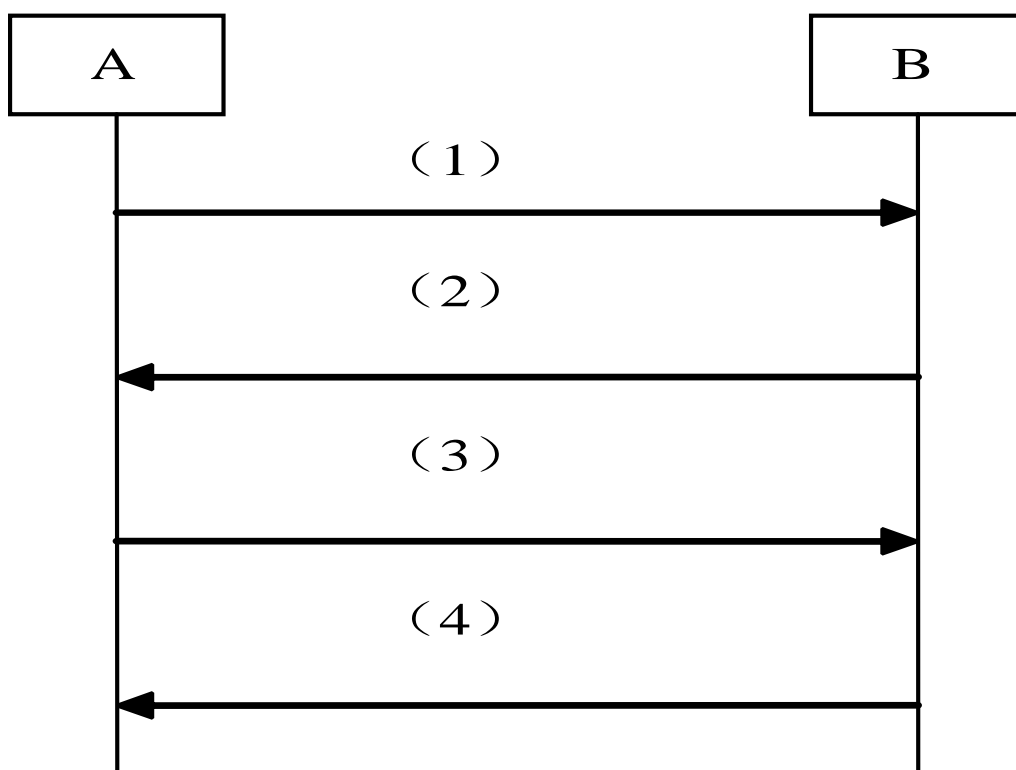
# 认证机制

- ◆ 采用认证和授权机制，可以对接入权限加以控制。
- ◆ 这里介绍四种SIP支持的认证机制：
  - n 端到端认证
  - n 代理认证（跳到跳认证）
  - n PGP认证
  - n HTTP的认证机制

# 端到端认证

被叫可在返回的401响应的WWW-AUTHENTICATION字段中给出其认可的认证体制和参数，然后主叫在请求的AUTHORIZATION字段给出包含特定消息头字段的数字签名，实现对SIP请求的发起者的身份认证。

# 端到端认证示意图



# 端到端认证流程

- ◆ 用户代理A向用户代理B发请求，请求中未含AUTHORIZATION字段。
- ◆ 用户代理B需要用户代理A进行授权认证，返回401（未授权）响应，并在响应的WWW-AUTHENTICATION字段中给出适合用户代理A的认证体制和参数。
- ◆ 用户代理A重发请求给用户代理B，在请求的AUTHORIZATION字段给出信任书，包含认证信息。
- ◆ 用户代理B收到认证请求，检查出用户代理A身份合法，返回200OK响应，用户代理A通过身份认证。

# 代理认证（跳到跳认证）

- ◆代理认证方式用于代理服务器对用户代理或其他用户代理的认证，代理服务器在407响应中通过PROXY-AUTHENTICATE字段，给出其认证体制和参数；
- ◆用户代理可利用PROXY-AUTHENTICATE字段向需要身份认证的代理证实自己的身份。，流程同上，只是需要身份认证的代理服务器。

# PGP认证

- ◆ PGP认证机制基于如下模型：用户通过在请求中进行密钥签名向服务器认证身份，服务器根据公开密钥来判断请求的来源。
- ◆ 详细信息参见IETF的RFC 2015。

# HTTP认证

- ◆ SIP协议支持HTTP的“BASIC”和“DIGEST”机制。
- ◆ 详细信息参见IETF的RFC 2616。