



# SIP协议分析

## — SIP协议基础架构 (续)

李静林



## 理解协议处理的基本原理与实现模型

- 协议的结构
- 消息的处理规则
  
- 用户管理
- 组网与路由
- 会话建立与媒体协商



# 用户管理

- SIP/SIPS用户地址 — SIP/SIPS URI
  - SIP and SIPS Uniform Resource Indicators
  - 格式：  
**sip:user:password@host:port;uri-parameters?headers**
  - 合法的SIP URI
    - sip:alice@atlanta.com
    - sip:bob@biloxi.com:5060
    - sip:bob@biloxi.com;transport=udp
    - sip:bob@192.0.2.4
    - sip:bob@biloxi.com:5060;lr
  - 合法的URI
    - tel:+358-555-1234567;postd=pp22
    - sip:+358-555-1234567;postd=pp22@foo.com;user=phone
    - RFC2806 (tel: URLs for Telephone Calls)
    - RFC2396 (Uniform Resource Identifiers (URI))
    - 等等

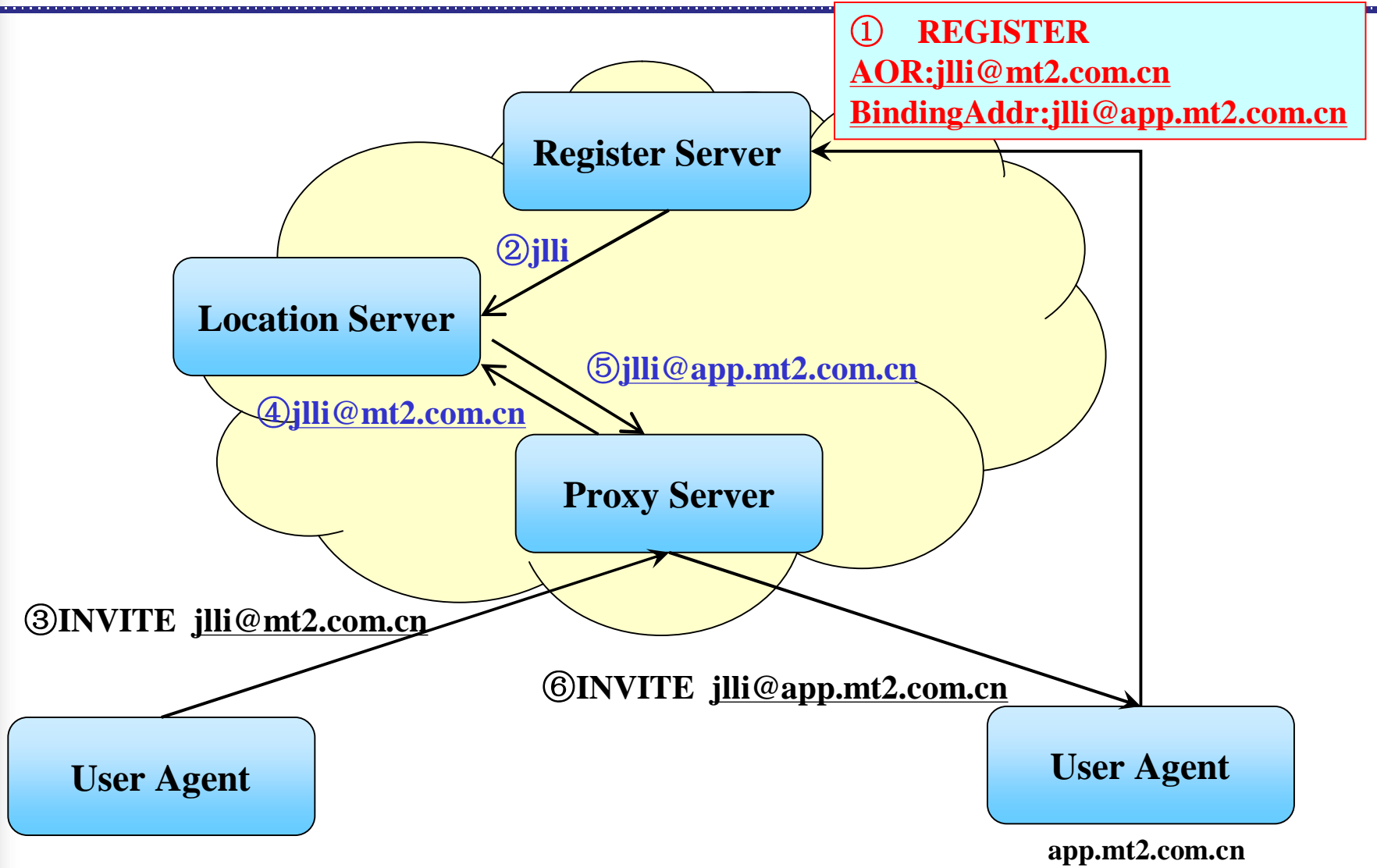


# 用户管理的基本概念

- Address-of-Record (AOR)
  - An address-of-record (AOR) is a **SIP or SIPS URI**
  - AOR points to a domain with a location service that can map the URI to another URI where the user might be available
  - Typically, the location service is populated through registrations
  - An AOR is frequently thought of as the "**public address**" of the user
- Home Domain (归属控制域)
  - The domain **providing service** to a SIP user
  - Typically, this is the domain present in the URI in the address-of-record (AOR) of a registration
- AOR — HOME Domain的关系
  - AOR是用户的唯一号码
  - HOME Domain维护用户唯一号码与实际号码的映射关系



# 注册与用户管理过程





# 注册请求与响应

注册到的Register Server

**REGISTER** sips:ss2.biloxi.example.com SIP/2.0  
 Via: SIP/2.0/TLS client.biloxi.example.com:5061;bran  
 Max-Forwards: 70  
**To:** Bob <sips:bob@biloxi.example.com>  
**From:** Bob <sips:bob@biloxi.example.com>;tag=ja743ks76z1f1H  
 Call-ID: 1j9FpLxk3uxtm8tn@biloxi.example.com  
 CSeq: 2 REGISTER  
**Contact:** <sips:bob@client.biloxi.example.com>  
 Content-Length: 0

要注册的AOR  
 必须是SIP / SIPS地址

发起注册请求的AOR  
 缺省From=To  
 特殊：第三方注册

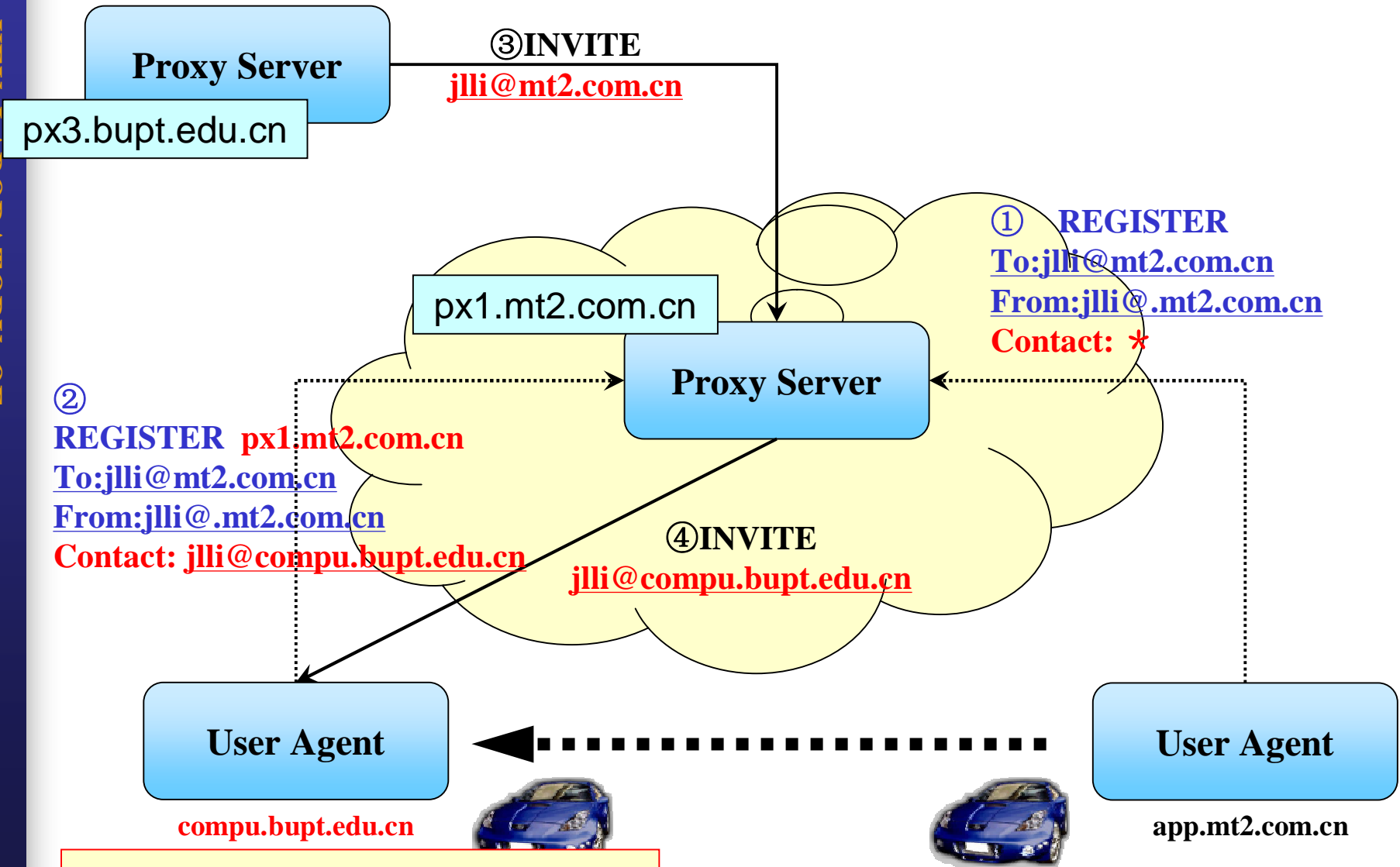
注册AOR所绑定的实际联系地址  
 可以携带0~n个合法的URL地址  
**Contact: \*** = 取消注册  
 不携带Contect = 查询

SIP/2.0 200 OK  
 Via: SIP/2.0/TLS client.biloxi.example.com:5061;received=192.0.2.201  
**From:** Bob <sips:bob@biloxi.example.com>;tag=ja743ks76z1f1H  
**To:** Bob <sips:bob@biloxi.example.com>;tag=37GkEh  
 Call-ID: 1j9FpLxk3uxtm8tn@biloxi.example.com  
 CSeq: 2 REGISTER  
**Contact:** <sips:bob@client.biloxi.example.com>;expires=3600  
**Contact:** <mailto:bob@biloxi.example.com>;expires=4294967295  
 Content-Length: 0

当前已绑定的联系地址  
 及相应的生存周期  
 取消注册：返回空



# Personal Mobility

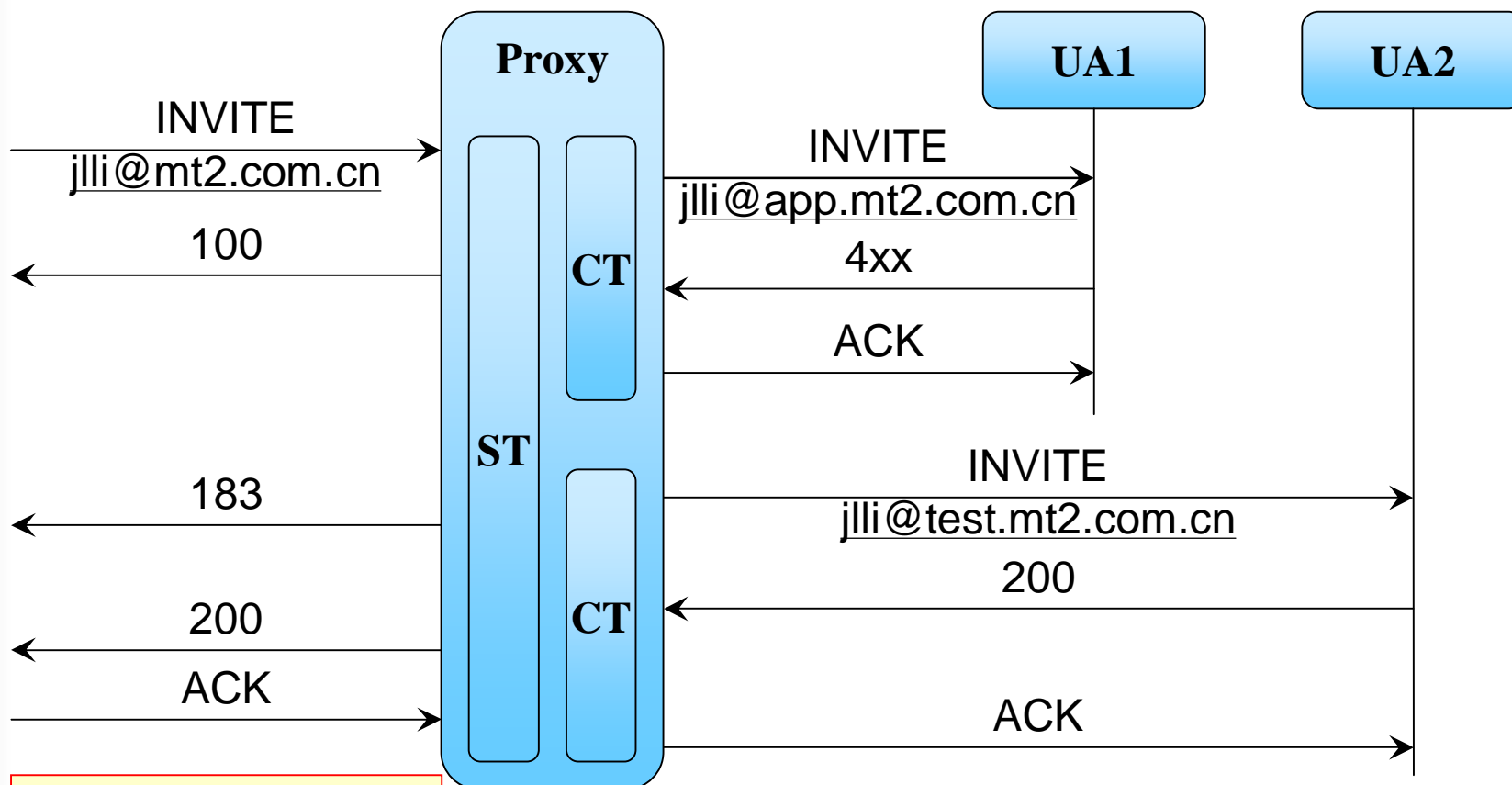


注意：移动到哪里？注册到哪里？



## 绑定多个有效地址

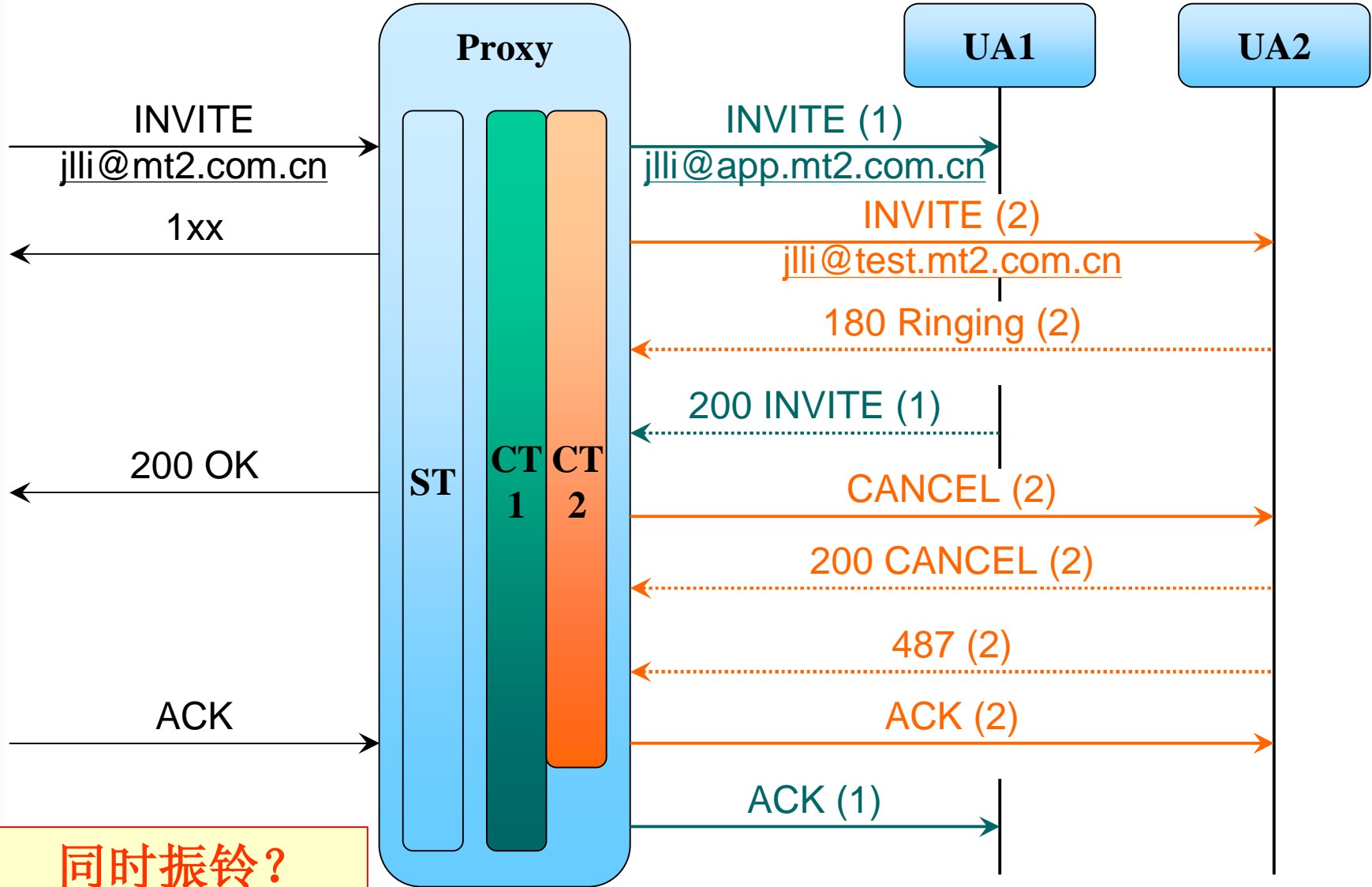
- AOR: [jlli@mt2.com.cn](mailto:jlli@mt2.com.cn)
  - Binding Address 1: [jlli@app.mt2.com.cn](mailto:jlli@app.mt2.com.cn)
  - Binding Address 2: [jlli@test.mt2.com.cn](mailto:jlli@test.mt2.com.cn)



依次振铃?



# Forking



同时振铃?



## 用户管理小结

- Domain
- AOR
- 注册请求与响应
- 注册过程
  
- 基于Target Set可以做的事情
  - Personal Mobility
  - 依次振铃
  - 同时振铃

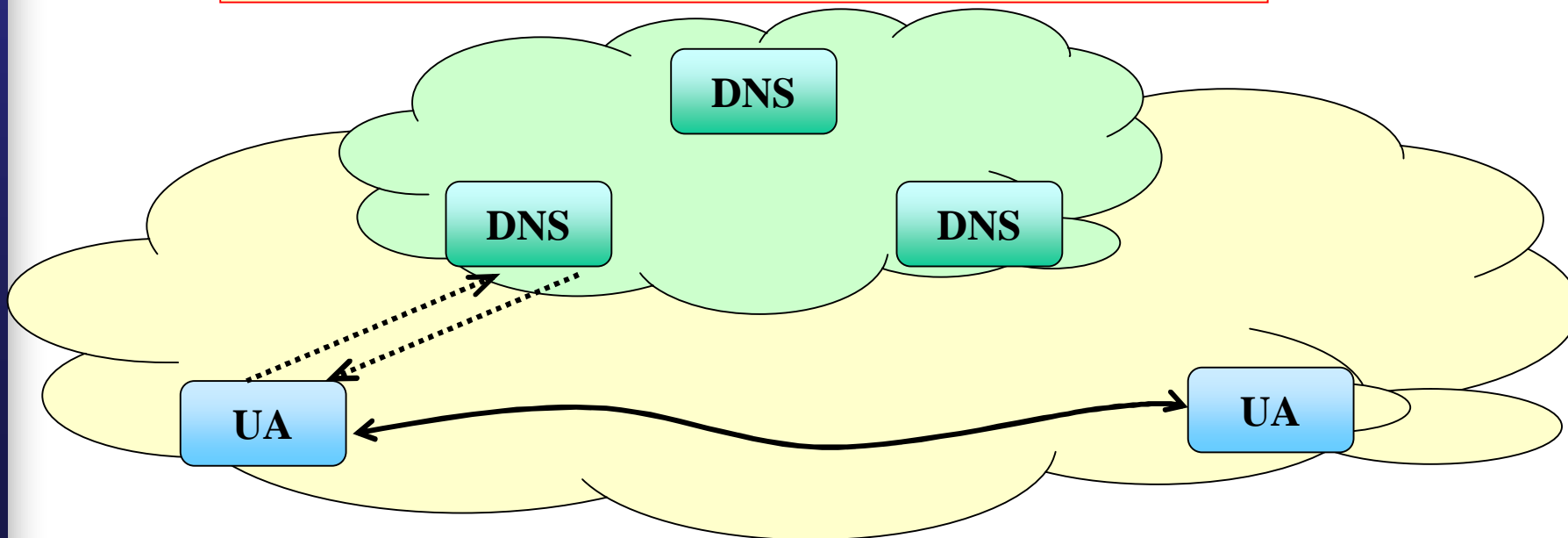
这些是**SIP**协议支持的业务能力吗？

# 组网与路由 Networking & Routing

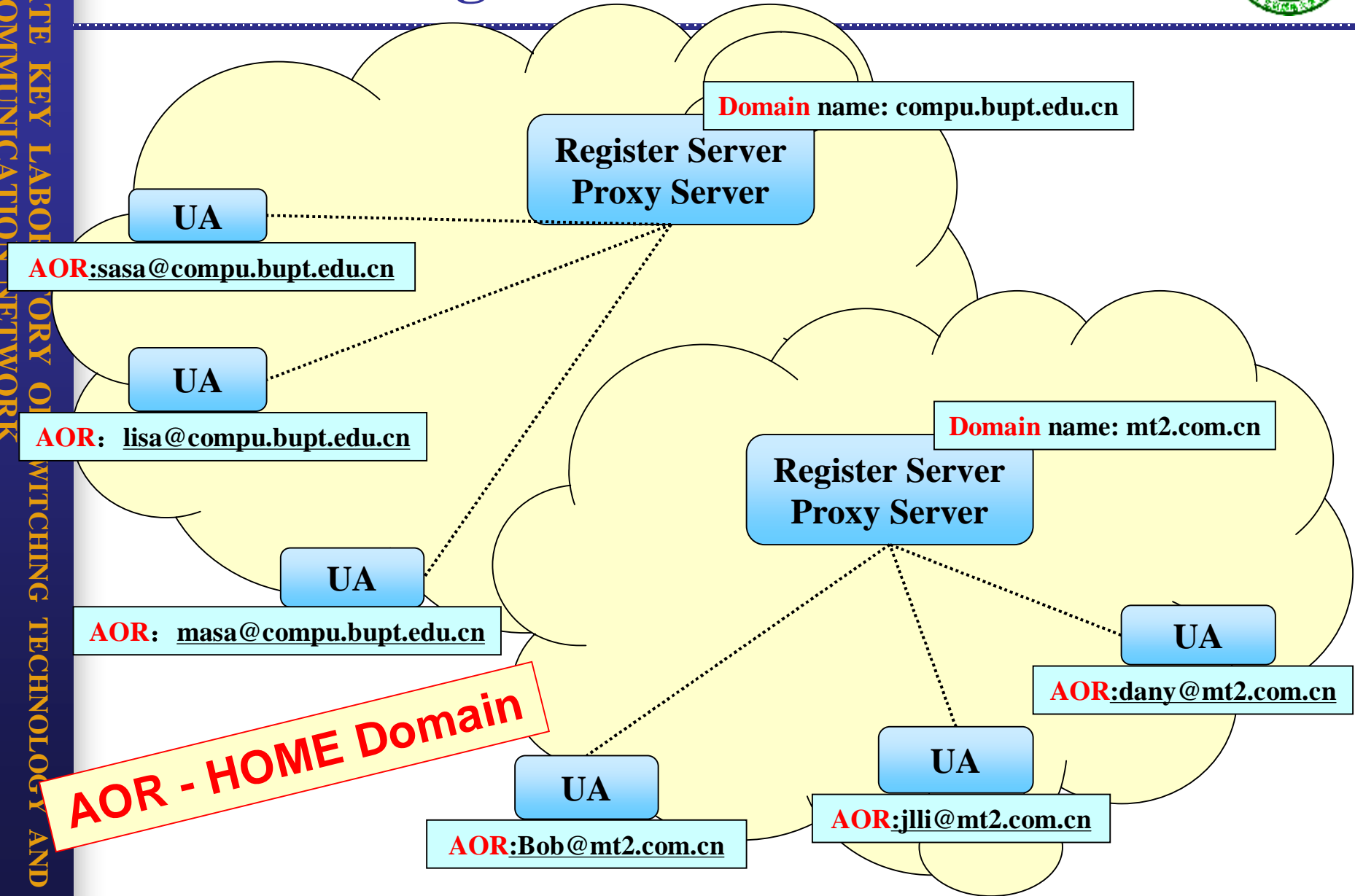


- Networking
  - SIP协议需要组网吗?
  - DNS解析被叫用户地址不可以吗?
  - 回想 SIP协议注册机制: AOR—HOME Domain

**SIP组网的目的是完成用户管理**

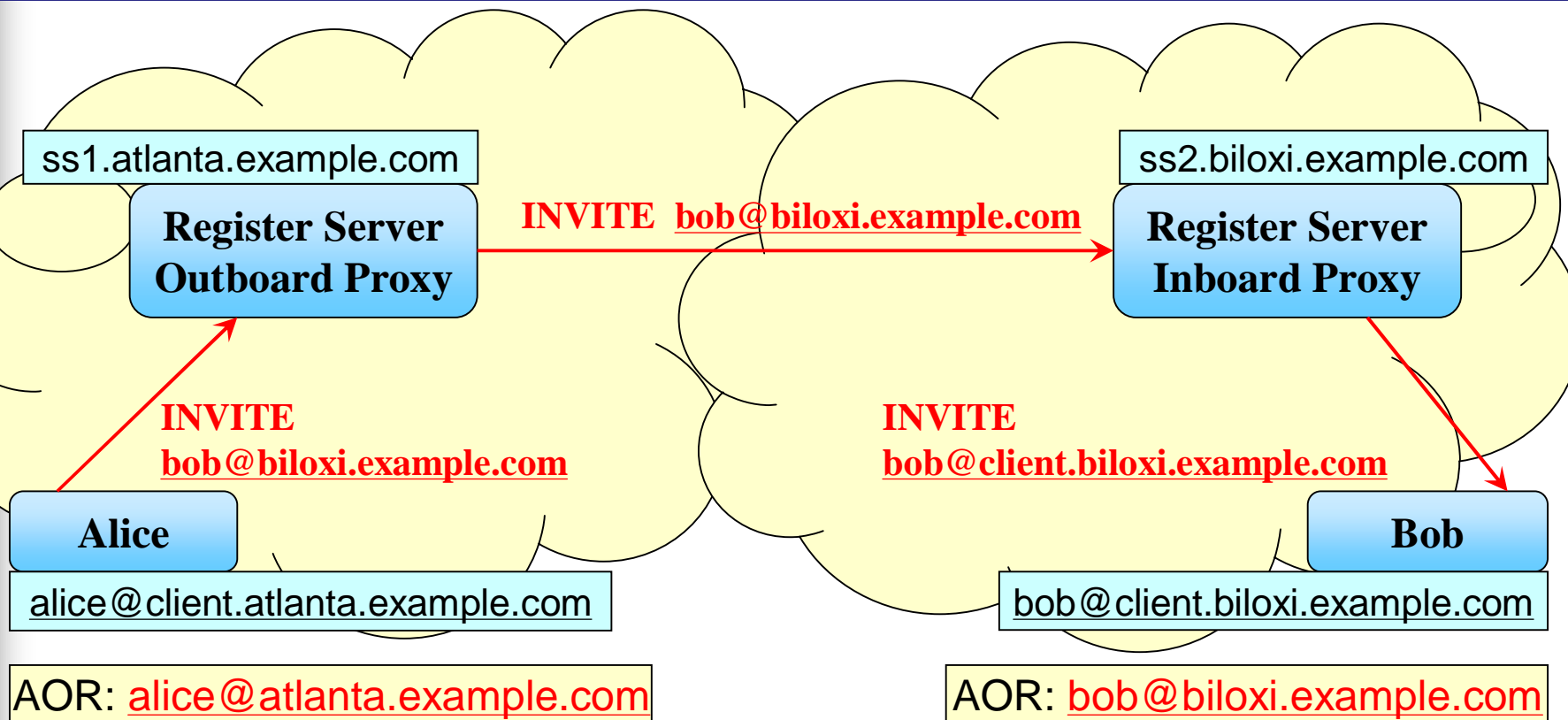


# Networking





# 路由过程



- 如何将请求路由到目的地
  - Alice-Proxy1: UAC如何转发请求
  - Proxy1-Proxy2: Proxy如何转发目的地非本控制域的请求
  - Proxy2-Bob: Proxy如何转发目的地为本控制域的请求



# 路由过程

↓ Alice-Proxy1

INVITE sip:bob@biloxi.example.com SIP/2.0  
 Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9;received=192.0.2.101  
 Max-Forwards: 70  
 Route: <sip:ss1.atlanta.example.com;lr>  
 Contact: <sip:alice@client.atlanta.example.com;transport=tcp>

Request-URI  
被叫AOR

Top Via主叫地址

指定Route  
URL中包含lr参数

主叫实际地址

↓ Proxy1-Proxy2

INVITE sip:bob@biloxi.example.com SIP/2.0  
 Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1;received=192.0.2.101  
 Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9;received=192.0.2.101  
 Max-Forwards: 69  
 Record-Route: <sip:ss1.atlanta.example.com;lr>  
 Contact: <sip:alice@client.atlanta.example.com;transport=tcp>

Request-URI不变

TopVia添加实际上一跳IP地址; 添加TopVia;

递减

删除Route,添加Record-Route

↓ Proxy2-Bob

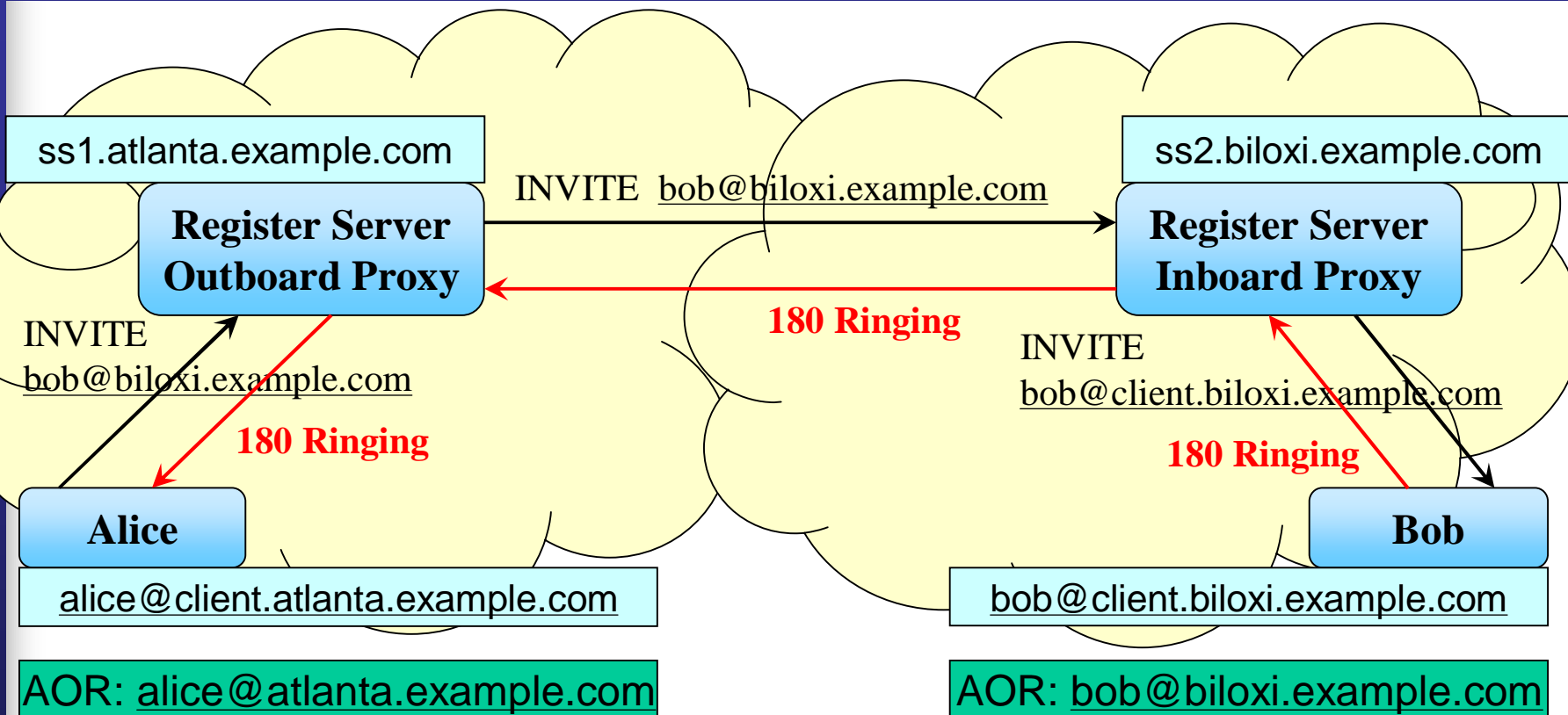
INVITE sip:bob@client.biloxi.example.com SIP/2.0  
 Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1  
 Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1;received=192.0.2.111  
 Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9;received=192.0.2.101  
 Max-Forwards: 68  
 Record-Route: <sip:ss2.biloxi.example.com;lr>,<sip:ss1.atlanta.example.com;lr>  
 Contact: <sip:alice@client.atlanta.example.com;transport=tcp>

Bob属于本控制域, Request-URI改变为AOR绑定地址

带到了终端Bob  
 实际转发路径 (Via)  
 主叫实际地址 (Contact)  
 强制路由 (Record-Route)



# 路由过程



- 如何将请求的响应返回到发端
  - Bob-Proxy2: UAS如何返回响应
  - Proxy2-Proxy1: Proxy如何转发响应
  - Proxy1-Alice: Proxy如何转发响应



# 路由过程

## ↓ Bob-Proxy2

SIP/2.0 200 OK

Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1;received=192.0.2.222

Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1;received=192.0.2.111

Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9;received=192.0.2.101

Record-Route: <sip:ss2.biloxi.example.com;lr>,<sip:ss1.atlanta.example.com;lr>

Contact: <sip:bob@client.biloxi.example.com;transport=tcp>

TopVia添加实际上一跳IP地址

## ↓ Proxy2-Proxy1

SIP/2.0 200 OK

Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1;received=192.0.2.111

Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9;received=192.0.2.101

Record-Route: <sip:ss2.biloxi.example.com;lr>,<sip:ss1.atlanta.example.com;lr>

Contact: <sip:bob@client.biloxi.example.com;transport=tcp>

Contact变为被叫实际地址

Record-Route不变

删除TopVia

## ↓ Proxy1-Alice

SIP/2.0 200 OK

Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9;received=192.0.2.101

Record-Route: <sip:ss2.biloxi.example.com;lr>,<sip:ss1.atlanta.example.com;lr>

Contact: <sip:bob@client.biloxi.example.com;transport=tcp>

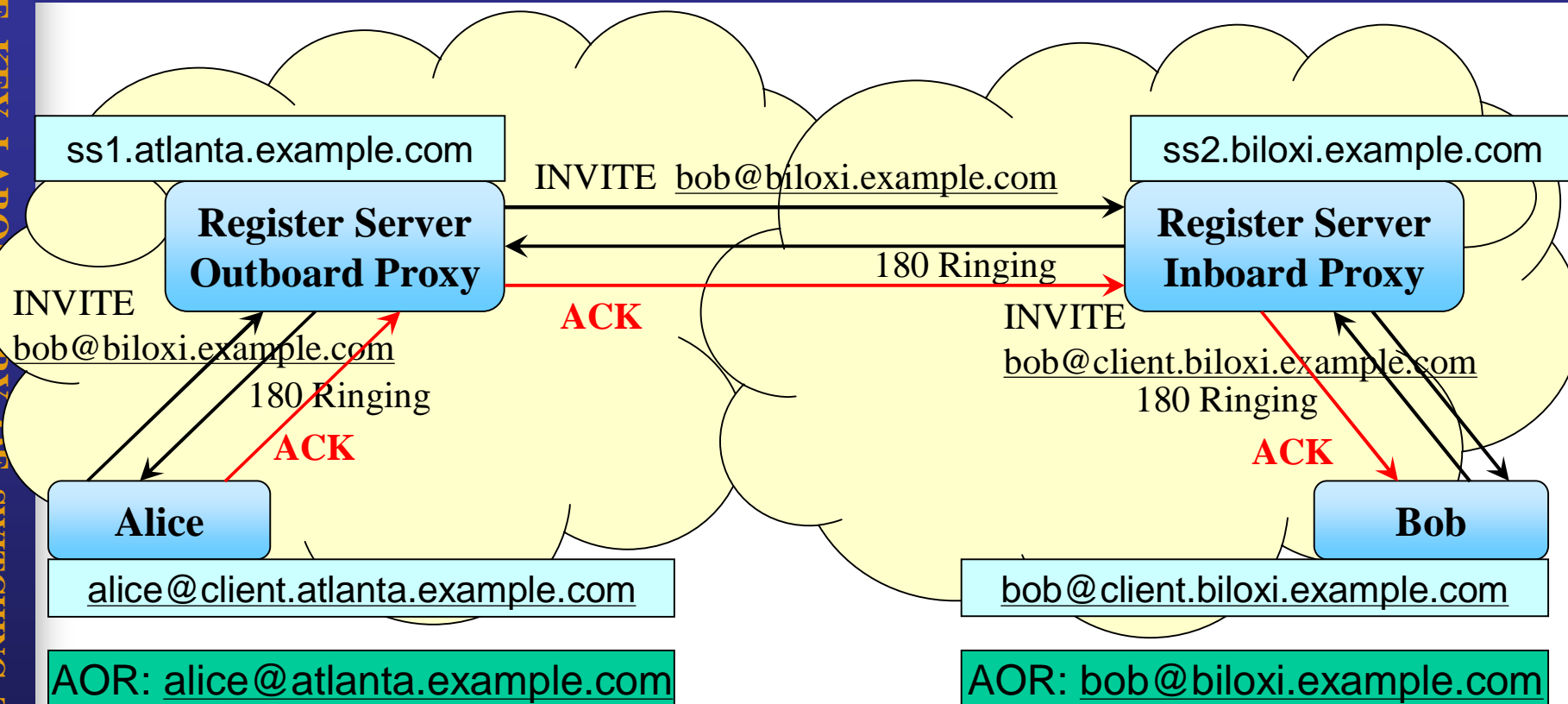
带到了终端Alice

被叫实际地址 (Contact)

强制路由 (Record-Route)



# 路由过程



- 如何转发后继请求
  - Alice-Proxy1: UAC如何处理后继请求
  - Proxy1-Proxy2: Proxy如何转发后继请求
  - Proxy2-Bob: Proxy如何转发目的地为本控制域的后继请求



# 路由过程

↓ Alice-Proxy1

Request-URI变为实际的被叫地址

ACK sip:bob@client.biloxi.example.com SIP/2.0  
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74b76  
Max-Forwards: 70  
Route: <sip:ss1.atlanta.example.com;lr>,<sip:ss2.biloxi.example.com;lr>

↓ Proxy1-Proxy2

根据Record-Route  
记录设置的Route

ACK sip:bob@client.biloxi.example.com SIP/2.0  
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1  
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74b76;received=192.0.2.101  
Max-Forwards: 69  
Route: <sip:ss2.biloxi.example.com;lr>

删除本Proxy的  
Route记录

↓ Proxy2-Bob

直接根据Request-URI  
就可以转发了

ACK sip:bob@client.biloxi.example.com SIP/2.0  
Via: SIP/2.0/TCP ss2.biloxi.example.com:5060;branch=z9hG4bK721e4.1  
Via: SIP/2.0/TCP ss1.atlanta.example.com:5060;branch=z9hG4bK2d4790.1;received=192.0.2.111  
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74b76;received=192.0.2.101  
Max-Forwards: 68

虽然Request-URI填写的是实际被叫地址，但是ACK将发送给  
**Top Route** 指定的地址：**强制路由**  
最后一跳，不需要再次进行地址查询就可以直接转发



## 路由过程总结

- 请求
  - 依据Route记录转发请求
  - 如果没有Route, 则依据Request-URI转发请求
  - 添加Record-Route记录, 将Proxy记录在后继请求转发路径中
  - 添加Via记录, 将Proxy记录在响应转发路径中
  - 路由跳数Max-Forward减一
- 响应
  - 依据Via转发响应
- 作用
  - Via保证了响应与请求的路径一致 (响应与请求的对应)
  - Record-Route / Route保证了关心这一Dialog的Proxy能够保留在后继请求的转发路径中
  - Max-Forward保证消息不会在网络中无限制的生存下去
  - Request-URI保证了请求目标不变



# 松散路由与严格路由

- 松散路由（Loose Routing）
  - A proxy is said to be loose routing if it follows the procedures defined in this specification for processing of the Route header field.
  - These procedures **separate** the destination of the request (present in the **Request-URI**) from the set of proxies that need to be visited along the way (present in the **Route** header field).
  - A proxy compliant to these mechanisms is also known as a loose router
- 严格路由（Strict Routing）
  - A proxy is said to be strict routing if it follows the Route processing rules of RFC 2543 and many prior work in progress versions of this RFC.
  - That rule caused proxies to **destroy** the contents of the **Request-URI** when a **Route** header field was **present**.
  - Strict routing behavior is not used in this specification, in favor of a loose routing behavior.
  - Proxies that perform strict routing are also known as strict routers.



# 松散路由与严格路由过程比较

松散路由

↓ alice@client.atlanta.example.com  
ss1.atlanta.example.com

严格路由

INVITE bob@biloxi.example.com  
To: bob@biloxi.example.com  
Route: ss1.atlanta.example.com

INVITE bob@biloxi.example.com  
To: bob@biloxi.example.com  
Route

接收端删除Route

发送端删除Route

↓ ss1.atlanta.example.com  
ss2.biloxi.example.com

INVITE bob@biloxi.example.com  
To: bob@biloxi.example.com  
Route-Request: ss1.atlanta.example.com

INVITE ss2.biloxi.example.com  
To: bob@biloxi.example.com  
Route-Request: ss1.atlanta.example.com

Request-URI不变

Request-URI修改为Proxy2

↓ ss2.biloxi.example.com  
bob@client.biloxi.example.com

INVITE bob@client.biloxi.example.com  
To: bob@biloxi.example.com  
Route-Request: ss2.biloxi.example.com  
Route-Request: ss1.atlanta.example.com

INVITE bob@client.biloxi.example.com  
To: bob@biloxi.example.com  
Route-Request: ss2.biloxi.example.com  
Route-Request: ss1.atlanta.example.com



# 松散路由与严格路由过程比较

松散路由

alice@client.atlanta.example.com  
↓  
ss1.atlanta.example.com

严格路由

ACK bob@client.biloxi.example.com  
To: bob@biloxi.example.com  
Route: ss1.atlanta.example.com  
Route: ss2.biloxi.example.com

ACK ss1.atlanta.example.com  
To: bob@biloxi.example.com  
Route  
Route: ss2.biloxi.example.com

Request-URI  
直接是Bob

ss1.atlanta.example.com  
↓  
ss2.biloxi.example.com

为了支持发送端删除  
Route, Request-URI修  
改为TopRoute

ACK bob@client.biloxi.example.com  
To: bob@biloxi.example.com  
Route: ss2.biloxi.example.com

ACK ss2.biloxi.example.com  
To: bob@biloxi.example.com  
Route

Request-URI不变

ss2.biloxi.example.com  
↓  
bob@client.biloxi.example.com

Request-URI修改为Proxy2

ACK bob@client.biloxi.example.com  
To: bob@biloxi.example.com

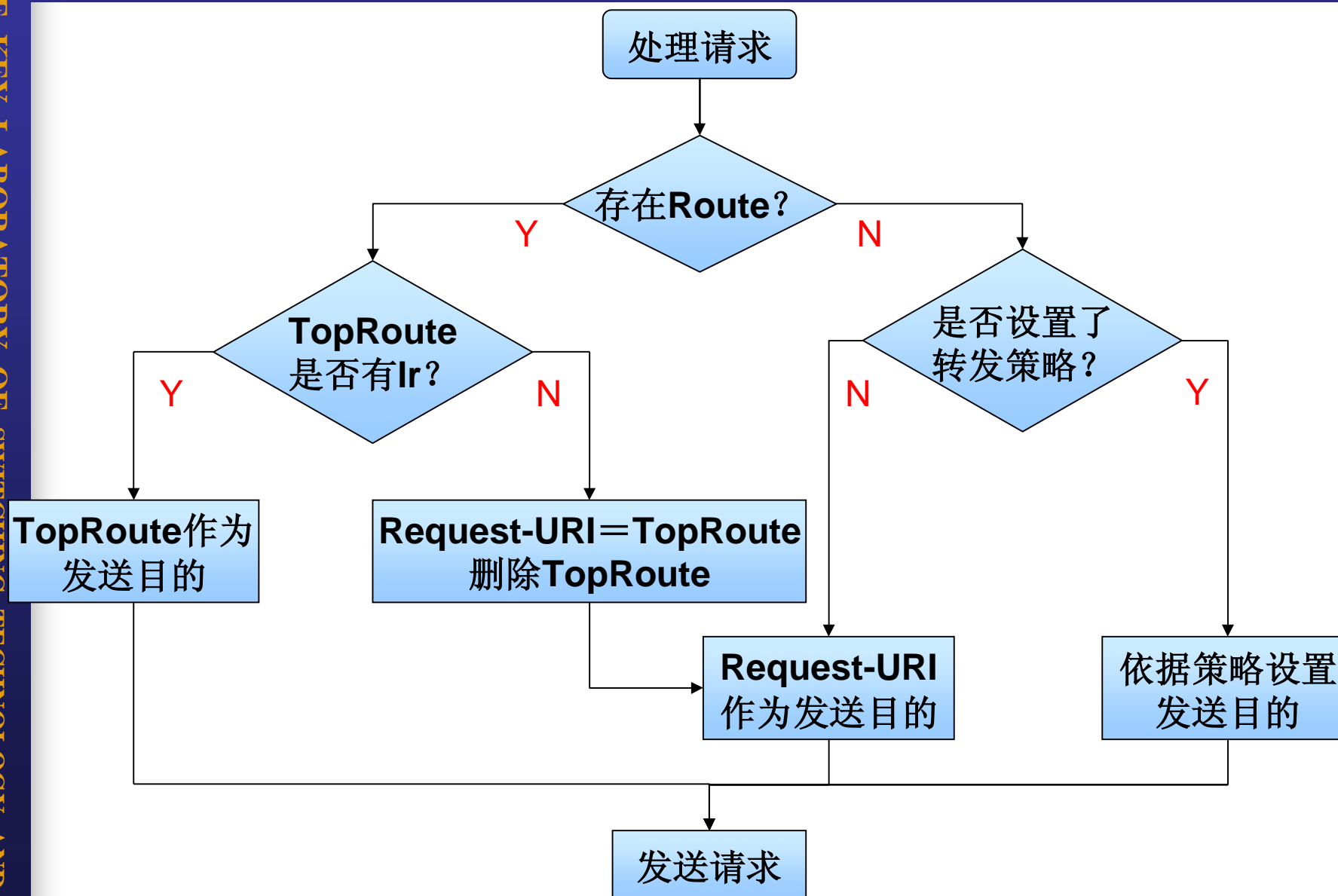
ACK bob@client.biloxi.example.com  
To: bob@biloxi.example.com



# 松散路由与严格路由的区别

- 松散路由
  - 如果请求中Route存在，则依据Route路由
  - 如果请求中Route不存在，则依据Request-URI路由
  - 在整个路由路径中，**Request-URI不变**，指向最终目的
  - 后继请求使用Contact作为Request-URI，到达最终目标控制域之后仍然可以依据Request-URI直接转发
- 严格路由
  - 请求依据Request-URI路由
  - 如果请求中存在Route，则使用Route修改Request-URI
  - 在整个路由路径中，**Request-URI指向下一跳地址**
  - 后继请求即便使用Contact作为Request-URI，在到达最终目标控制域的时候，依然只能使用To指示路由目标，可能还需要进行一次路由查询
- 松散路由的优点
  - 谁能最准确的确定目标？ **目标所属的控制域**
  - Request-URI = To? **Request-URI可能包含更丰富的信息**
  - 实际From由 Client请求中的Contact 标识
  - 实际To由 Server最终响应中的Contact标识
  - From-To-CallID演变为 Dialog标识

## 兼容lr与sr的请求消息路由处理过程





## 组网与路由小结

- 组网
- 路由过程
- 松散路由
- 严格路由
- 兼容松散路由与严格路由

严格路由 — B2BUA ?



# 会话建立与媒体协商

- 会话建立的两层含义
  - 实体之间**逻辑连接关系**的建立
  - 实体之间**媒体流**的建立
- SIP协议完成的是“会话”建立吗
  - SIP协议完成的是：维持Peer-Peer之间的**逻辑连接关系**
  - SIP协议本身**不关心媒体流**的建立
- SIP协议“族”中有媒体建立与协商协议吗？
  - SDP: Session Description Protocol
    - SDP is intended for **describing multimedia sessions** for the purposes of session announcement, session invitation, and other forms of multimedia session initiation.
    - SDP完成的仅仅是**如何描述Multimedia Sessions**

**SIP协议是一个满足会话控制与媒体控制分离的协议吗**



# H.323协议的媒体控制

- **H.245 — 媒体控制协议**
- **H.245 session:**
  - The part of the call that begins with the establishment of an H.245 Control Channel and ends with the receipt of the H.245 EndSessionCommand or termination due to failure.
  - Not to be confused with a call, which is delineated by the H.225.0 Setup and Release Complete messages.
- **H.245的主要目的是**
  - Master/slave determination 主从确定
  - Capability Exchange 能力交换
  - Logical Channel Signalling 逻辑通道信令
  - Bidirectional Logical Channel Signalling
  - Close Logical Channel Signalling
  - Mode Request
  - Round Trip Delay Determination
  - Maintenance Loop Signalling



# IP网络上需要媒体控制协议吗

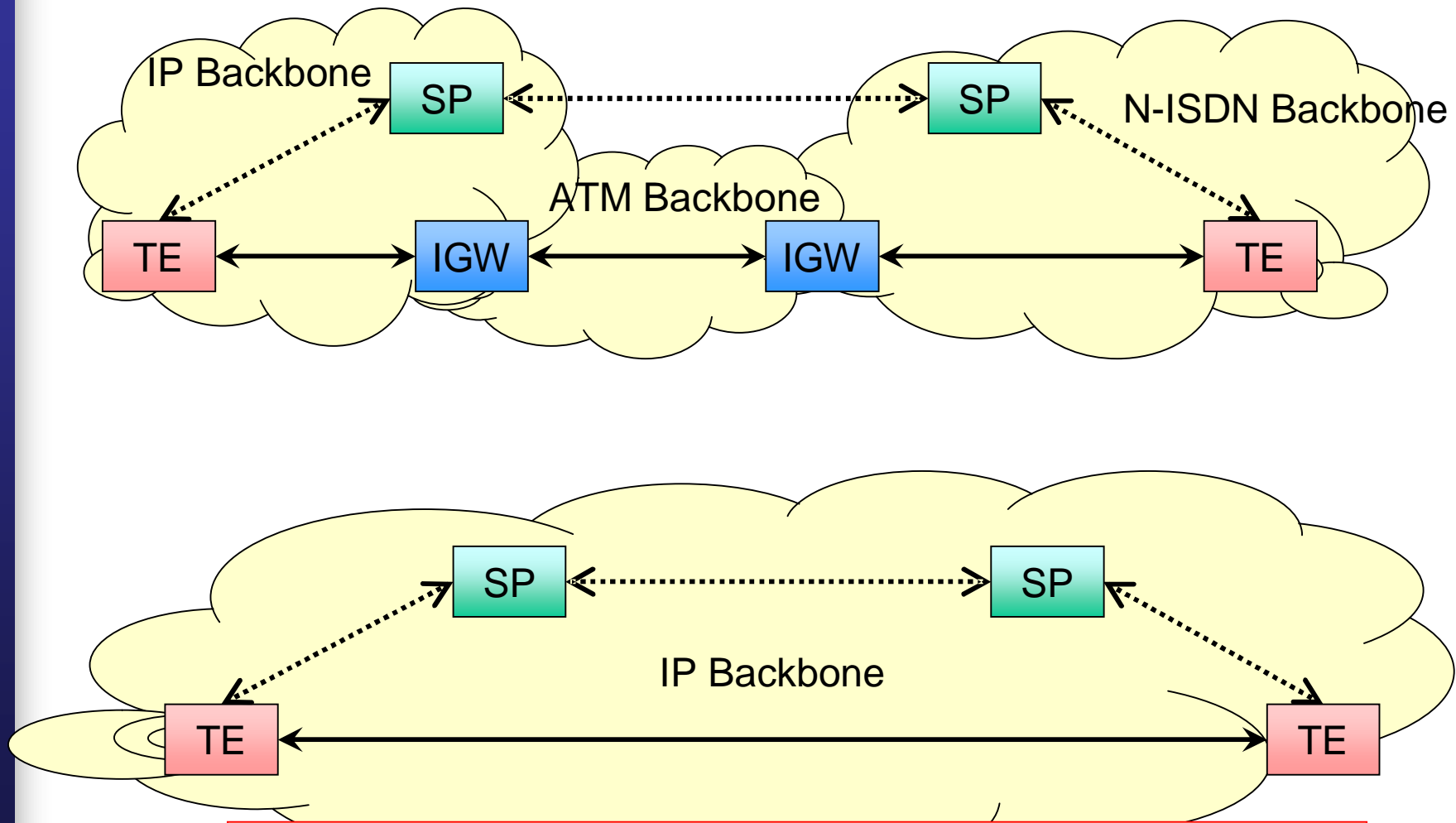
- IP网络上需要媒体控制协议吗
  - IP网络连接并进行数据传输的基本要求
    - TCP/UDP+IP地址+端口号 — 主从决定
    - 媒体类型描述与协商 — 能力交换
  - IP网络连接并进行数据传输的扩展要求
    - 实时媒体传输与控制 — RTP/RTCP
- H.323基于IP网络为什么有媒体控制协议
  - 在ITU-T的场景下，要适应广义的网络
  - H.3xx系列协议并不仅仅针对IP网络
  - H.245也不仅仅针对IP网络的媒体连接控制

需要吗?

需要吗?



# 媒体控制协议与媒体控制能力



媒体能力的协商语义 走信令通道还是走媒体通道还有区别吗

# SDP



- Session name and purpose
- Time(s) the session is active
  - An arbitrary list of start and stop times bounding the session
  - For each bound, repeat times such as "every Wednesday at 10am for one hour"
- The media comprising the session
  - The type of media (video, audio, etc)
  - The transport protocol (RTP/UDP/IP, H.320, etc)
  - The format of the media (H.261 video, MPEG video, etc)
  - Multicast
    - Multicast address for media
    - Transport Port for media
  - Unicast
    - Remote address for media
    - Transport port for contact address
- Information to receive those media (addresses, ports, formats and so on)
- Information about the bandwidth to be used by the conference
- Contact information for the person responsible for the session



# SDP描述

- Session description
  - v= (protocol version)
  - o= (owner/creator and session identifier).
    - o=<username> <session id> <version> <network type> <address type> <address> s= (session name)
  - s= (session name)
  - c= (connection information - not required if included in all media)
    - c=<network type> <address type> <connection address>
- Time description
  - t= (time the session is active)
    - t=<start time> <stop time>
- Media description
  - m= (media name and transport address)
    - m=<media> <port> <transport> <fmt list>
    - m=<media> <port>/<number of ports> <transport> <fmt list>
  - a=\* (zero or more media attribute lines)
    - a=recvonly / sendrecv / sendonly
    - a=fmtp:<format> <format specific parameters>

# SDP与Offer-Answer模型



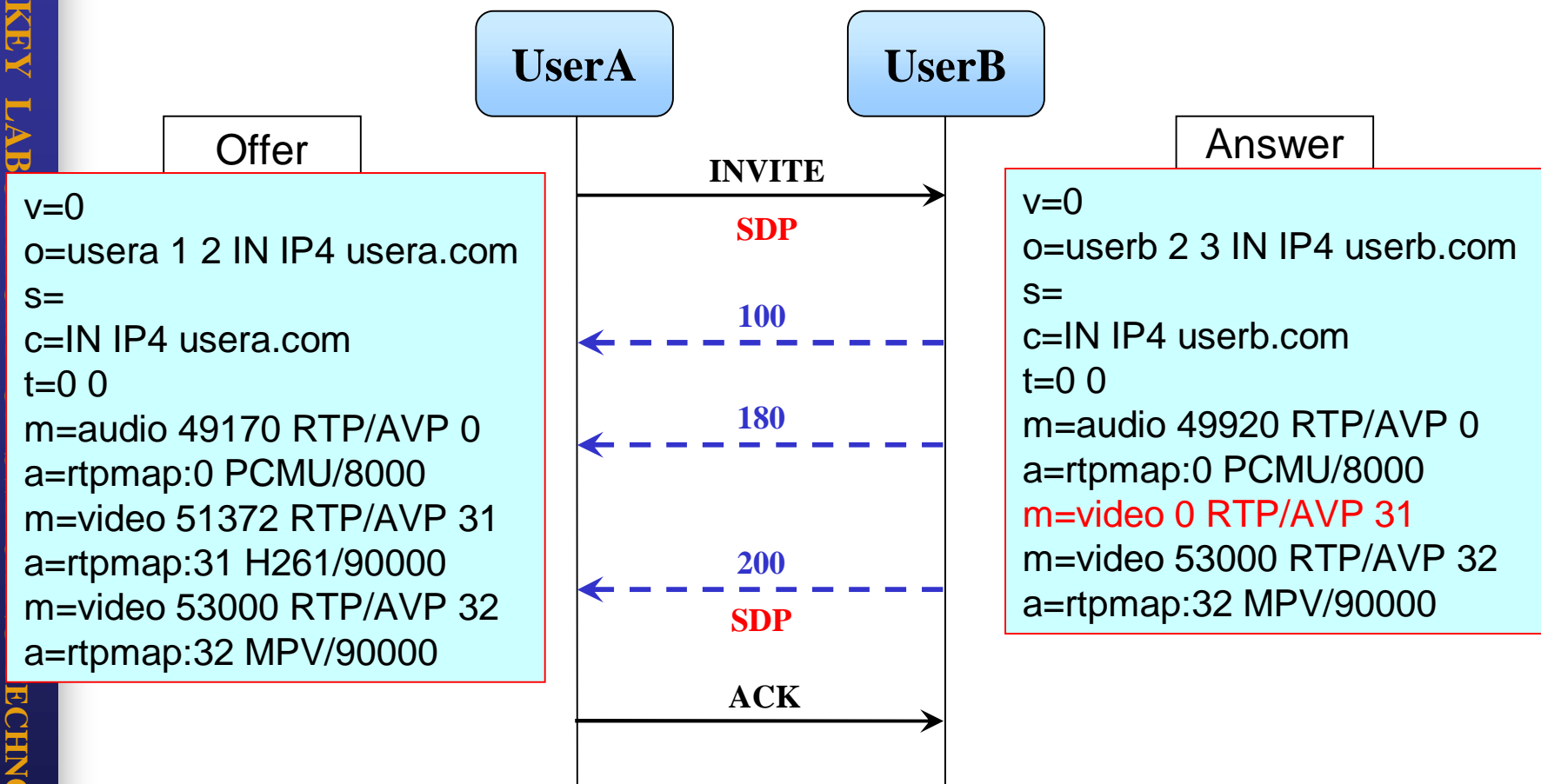
```
v=0
o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVP 31
a=rtpmap:31 H261/90000
m=video 53000 RTP/AVP 32
a=rtpmap:32 MPV/90000
```

Offer

```
v=0
o=bob 2890844527 2890844527 IN IP4 client.biloxi.example.com
s=-
c=IN IP4 192.0.2.201
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
```

Answer

# SIP协议与Offer-Answer模型



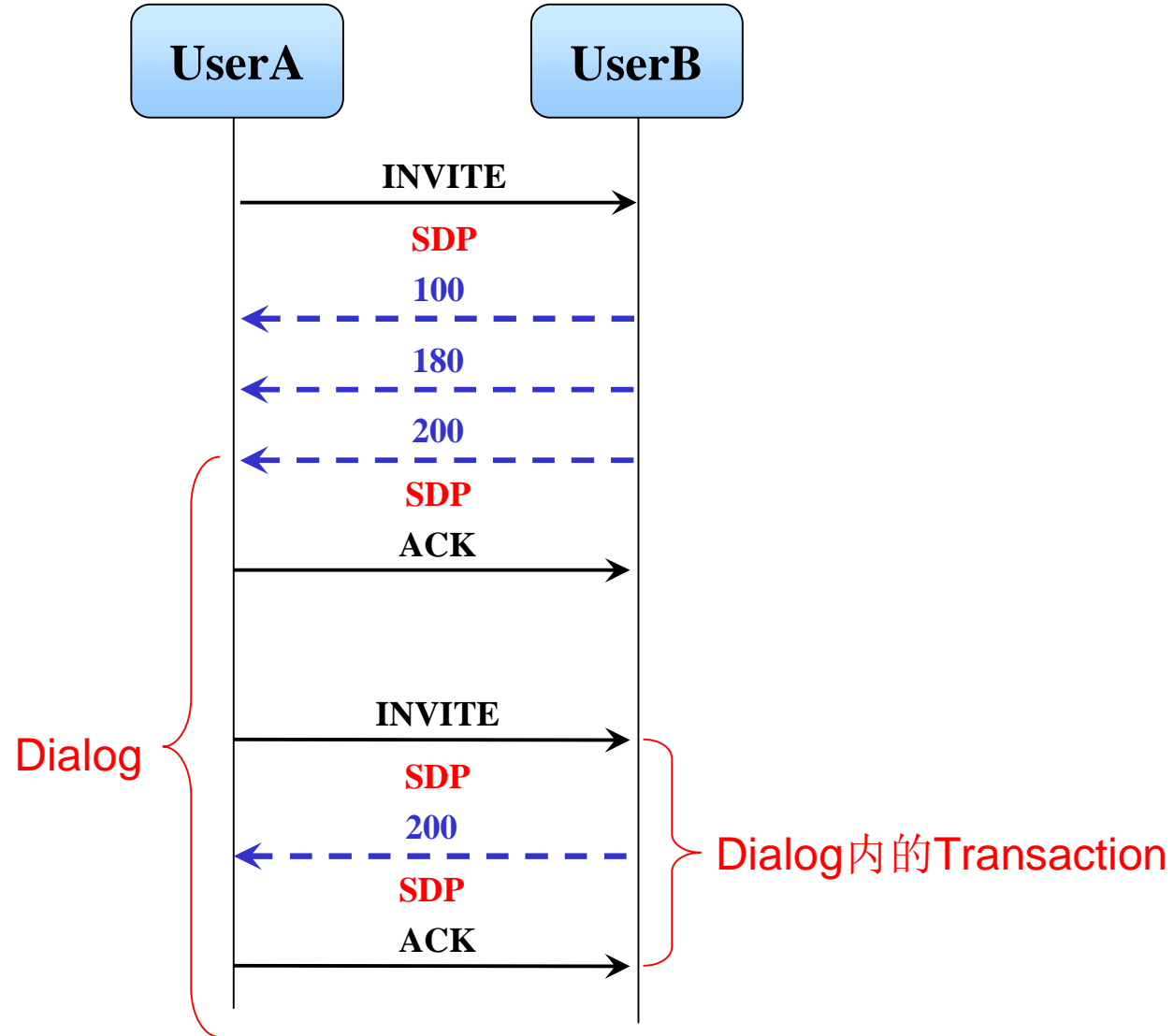
SDP是作为净荷承载的，SIP并不关心SDP所要表达的内容

SDP是呈现给处理SDP的终端 / 网关的，SIP并不关心Offer-Answer的过程

SDP一定是承载在SIP的可靠消息中的



# 会话过程中的媒体修改





## 会话建立与媒体协商小结

- SIP不是媒体建立 / 协商协议
- SDP的定义
- SDP如何描述会话
- Offer-Answer过程
- SDP与SIP的关系