



# MGCP协议分析

## - MGCP协议基本分析

# 媒体网关控制协议的需求



## RFC2805 - Media Gateway Control Protocol Architecture and Requirements

- Per-Call Requirements
  - Resource Reservation
  - Connection Requirements
  - Media Transformations
  - Signal/Event Processing and Scripting
  - QoS/CoS
  - Test Support
  - Accounting
  - Signalling Control
- Resource Control
  - Resource Status Management
  - Resource Assignment
- Operational/Management Requirements
  - Assurance of Control/Connectivity
  - Error Control
  - MIB Requirements
- General Protocol Requirements
  - MG-MGC Association Requirements
  - Performance Requirements
- Transport
  - Assumptions made for underlying network
  - Transport Requirements
- Security Requirements
- Requirements specific to particular bearer types
  - Media-specific Bearer Types
  - Application-Specific Requirements

# MGCP协议基本操作 – MGCP协议请求



- MG注册命令
  - **RSIP** RestartInProgress MG接入服务/退出服务
- 端点管理与查询
  - **EPCF** EndpointConfiguratin 对端点进行配置
  - **AUEP** AauditEndpoint 检测MG中特定端点的状态
  - **AUCX** AauditConnection 检测MG中特定连接的状态
- 事件监测设置与通知
  - **RQNT** NotificationRequest 请求MG处理信号,监测事件
  - **NTFY** Notify MG检测到了特定事件,并上报给MGC
- 连接管理
  - **CRCX** CreateConnection 在MG中创建特定的连接
  - **MDCX** ModifyConnection 修改MG中连接的属性
  - **DLCX** DeleteConnection 删除MG中建立的连接

MGCP所有的请求都针对“端点”

# MGCP协议基本操作 – MGCP协议响应



- 1xx 临时的响应
  - 100 事务正被执行
- 2xx 命令成功完成
  - 200 请求的事务成功完成
  - 250 连接成功删除
- 4xx 暂时性错误
  - 400 事务由于暂时性错误不能被执行
  - 403 端点当前没有足够的资源
  - 404 目前带宽不足
- 5xx 永久性错误
  - 500 事务由于端点未知而不能执行
  - 501 事务由于端点未准备好而不能执行
  - .....

MGCP是基于“请求—响应”的协议

MGCP是基于“Transaction”的协议



# MGCP的消息结构

命令行
参数行
会话描述

generic-message = Command Line  
 \*Parameter Line  
 [Session Description]  
 Command Line =RequestMethod **TransactionID** **EndpointName** ProtocolVersion  
 Parameter Line =ParameterName : [WS]ParameterValue

<b>NotificationRequest (RQNT)</b>	
[RequestedEvents]	<b>R</b>
RequestIdentifier	<b>X</b>
[DigitMap]	<b>D</b>
[SignalRequests]	<b>S</b>
[DetectEvents]	<b>T</b>
[NotifiedEntity]	<b>N</b>

<b>Notify (NTFY)</b>	
RequestIdentifier	<b>X</b>
ObservedEvents	<b>O</b>
[NotifiedEntity]	<b>N</b>

<b>ModifyConnection (MDCX)</b>	
CallID	<b>C</b>
ConnectionID	<b>I</b>
[NotifiedEntity]	<b>N</b>
[LocalConnectionOptions]	<b>L</b>
[ConnectionMode]	<b>M</b>
[RemoteConnectionDescriptor]	<b>RC</b>

<b>CreateConnection (CRCX)</b>	
CallID	<b>C</b>
[NotifiedEntity]	<b>N</b>
[LocalConnectionOptions]	<b>L</b>
ConnectionMode	<b>M</b>
[{RemoteConnectionDescriptor  SecondEndpointId}]	<b>RC Z2</b>

<b>RestartInProgress (RSIP)</b>	
RestartMethod	<b>RM</b>
[RestartDelay]	<b>RD</b>
[ReasonCode]	<b>E</b>

<b>DeleteConnection (DLCX)</b>	
CallID	<b>C</b>
ConnectionID	<b>I</b>



# MGCP协议扩展包

Generic Media Package			
Package Name: <b>G</b>			
Symbol	Definition	R	S
rt	Ringback Tone		TO

DTMF Package			
Package Name: <b>D</b>			
Symbol	Definition	R	S
1	DTMF 1	✓	BR
#	DTMF #	✓	BR
*	DTMF *	✓	BR
T	Interdigit Timer	✓	TO

Line Package			
Package Name: <b>L</b>			
Symbol	Definition	R	S
bz	Busy Tone		TO
dl	Dial Tone		TO
rg	Ringing		TO
hd	On-hook Transition	S	
hu	Off-hook Transition	S	

Trunk Package			
Package Name: <b>T</b>			
Symbol	Definition	R	S
bl	Blocking		BR
lb	Loopback		OO

R	Events	OO	On/Off signal	BF	Brief signal
S	Signals	TO	Time-Out signal	✓	can be requested

# MGCP消息举例



**RQNT** 2050 [aaln/1@rgw.com.cn](mailto:aaln/1@rgw.com.cn) mgcp 1.0

**R:** L/hu(n), D/[0-9#\*T](D)

**S:** L/dl

**X:** 2222

**D:** (0T|00T|#xxxxxxx|\*xx|91x|9011x.T)

200 2050 ok

**NTFY** 25 [aaln/1@rgw.com.cn](mailto:aaln/1@rgw.com.cn) mgcp 1.0

**O:** D/9, D/1, D/1

**X:** 2222

200 25 ok

**CRCX** 2052 [aaln/1@rgw.com.cn](mailto:aaln/1@rgw.com.cn) mgcp 1.0

**C:** 9876543210

**L:** p:20,a:PCMU

**M:** sendrecv

v=0

o=1234 4321 IN IP4 100.101.102.103

s=-

t=0 0

c=IN IP4 100.101.102.103

m=audio 49170 RTP/AVP 0

a=rtpmap:0 PCMU/8000

**200** 2052 ok

**I:** 0123456789

v=0

o=5678 8765 IN IP4 100.101.102.104

s=-

t=0 0

c=IN IP4 100.101.102.104

m=audio 49170 RTP/AVP 0

a=rtpmap:0 PCMU/8000

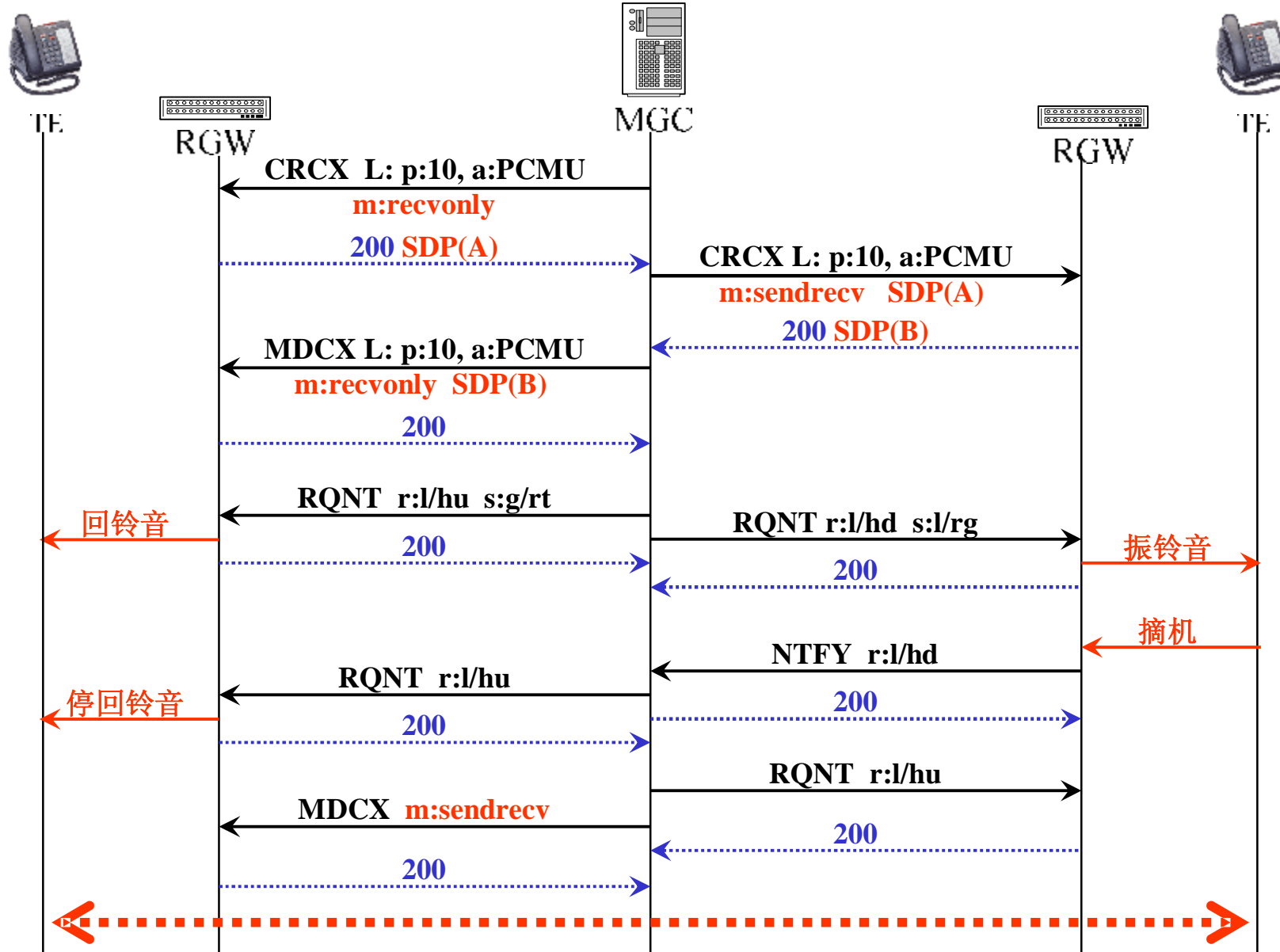


# MGCP协议基本网关控制流程





# MGCP协议基本网关控制流程







# MGCP协议操作总结

- 以“Endpoint”为核心的网关管理
- 以“Connection”为核心的Endpoint管理
  - 以Connection Identifier对应一组“CRCX-MDCX-DLCX”
  - 以Call Identifier维护一组Connection
- 以“Transaction”为基本消息处理单位
- 以Signal和Event为基本控制元素
  - 以Request Identifier对应 RQNT—NTFY
- 以Package完成功能组织



# Megaco协议分析

- Megaco协议控制模型  
与基本协议分析

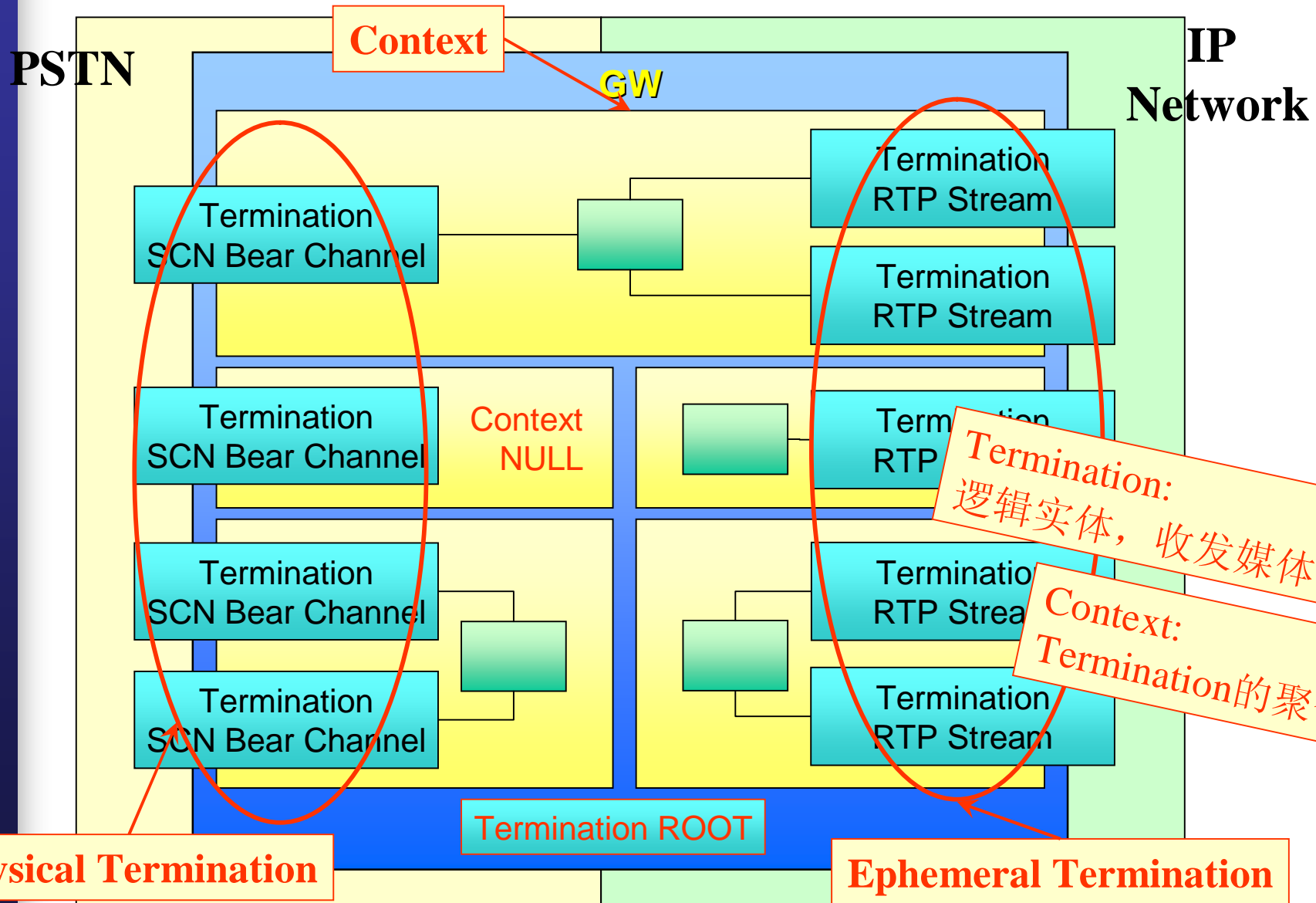


# Megaco协议控制模型基本定义

- Connection Model
  - The connection model for the protocol describes the **logical entities**, or **objects**, within the Media Gateway that can be controlled by the Media Gateway Controller.
  - The main abstractions used in the connection model are **Terminations** and **Contexts**.
- Termination
  - A Termination is a **logical entity** on a MG that sources and/or sinks media and/or control streams.
  - In a multimedia conference, a Termination can be multimedia and sources or sinks **multiple media streams**.
  - The media stream parameters, as well as modem, and bearer parameters are **encapsulated within the Termination**.
- Context
  - A Context is an **association** between **a collection of Terminations**.
  - The Context describes the **topology** (who hears/sees whom) and the media mixing and/or switching parameters if more than two Terminations are involved in the association.
  - There is a special type of Context, the **null** Context, which contains all Terminations that are not associated to any other Termination.
  - For instance, in a decomposed access gateway, **all idle lines are represented by Terminations in the null Context**.



# Megaco协议控制模型



Physical Termination

Ephemeral Termination

Termination:  
逻辑实体, 收发媒体流

Context:  
Termination的聚合





# Megaco协议基本操作

## Termination管理

<b>ServiceChange</b>	一个 / 一组Termination 加入 / 退出服务
<b>AuditValue</b>	获取Termination当前设置的特性、事件、信号、统计信息
<b>AuditCapabilities</b>	获取Termination可能的特性、事件、信号、统计信息

## Termination/Context操作

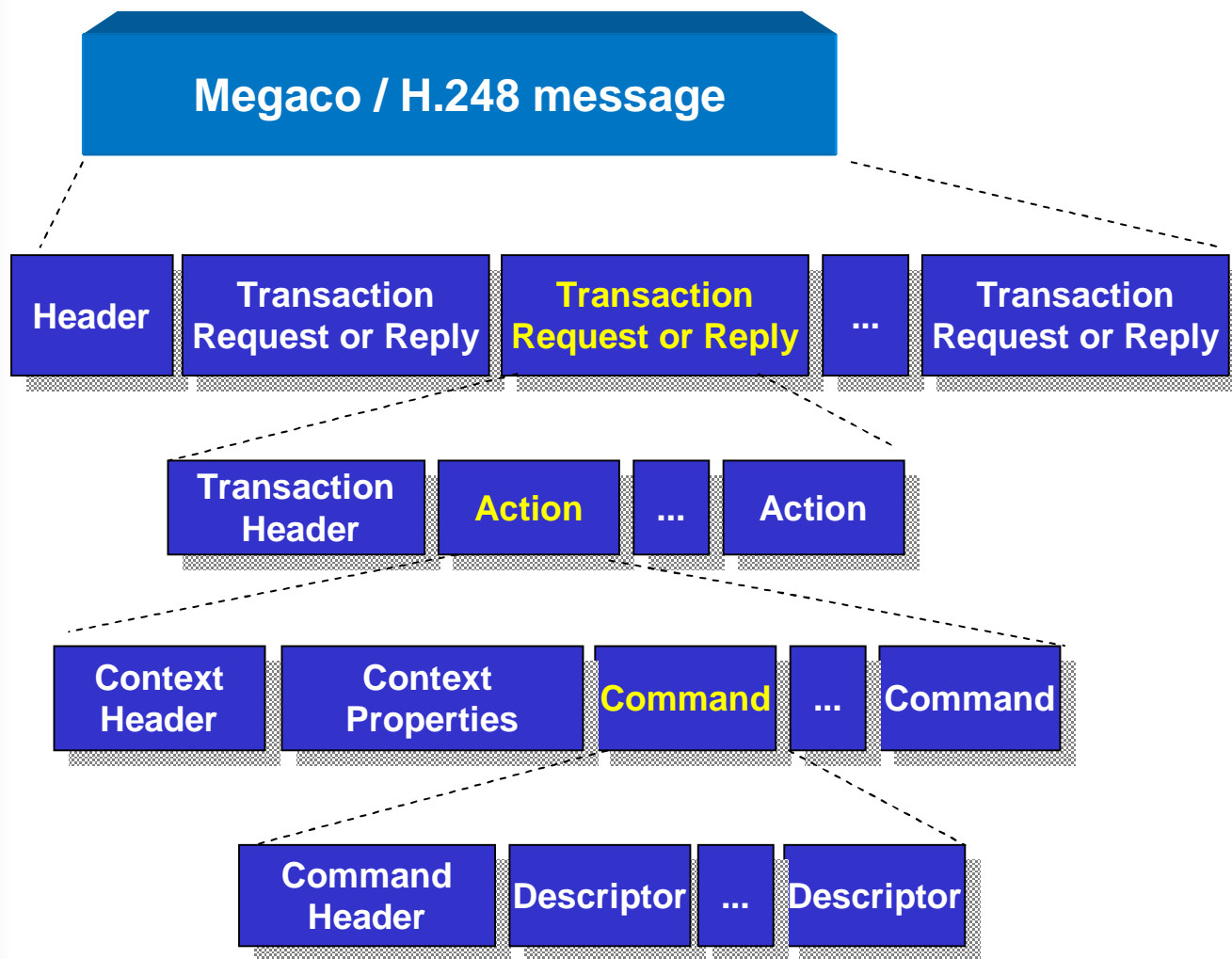
<b>Add</b>	向一个Context添加一个Termination
<b>Modify</b>	修改一个Termination的特性、事件和信号
<b>Subtract</b>	解除一个Termination与Context之间的联系
<b>Move</b>	Termination在Context之间的移动

## 事件通知

<b>Notify</b>	上报MG中发生的事件
---------------	------------



# Megaco协议的 消息结构



一个消息由一  
系列事务组成

一个事务由一  
系列动作组成

一个动作由操作  
一个Context的一  
系列命令组成

一个命令由一  
系列描述符组成



# Megaco的消息处理

MEGACO/1 [216.33.33.61]: 27000

```

Transaction = 1236 {
  Context = $ {
    Add = TermA {
      }
    Add = $ {
      Media {
        {
          LocalControl {
            Mode = Receiveonly,
          },
          Local {
            v=0
            c=IN IP4 $
            m=audio $ RTP/AVP 4
          }
        }
      }
    }
  }
}

```

MEGACO/1 [209.110.59.34]: 25000

```

Reply = 1236 {
  Context = 1 {
    Add = TermA,
    Add=EphA{
      Media {
        Local {
          v=0
          o=- 2890844526 2890842807 IN IP4 209.110.59.34
          s=-
          t= 00
          c=IN IP4 209.110.59.33
          m=audio 30000 RTP/AVP 4
          a=recvonly
        }
      }
    }
  }
}

```

# Megaco的消息处理



MEGACO/1 [216.33.33.61]: 27000

**Transaction** = 1248 {

**Context** = 1

**Topology** TermA, EphA,bothway, TermA, EphD, bothway, EphA,EphD, bothway{

**Modify** = EphA {

**Media** {

{

**LocalControl** {

Mode = sendrecv,

},

**Remote** {

v=0

c=IN IP4 192.168.0.110

m=audio 45000 RTP/AVP 4

}

}

}

}

}

MEGACO/1 [209.110.59.34]: 25000

**Reply** = 1248 {

**Context** = 1 {

**Modify**=EphA

}

}



# Megaco的消息处理

MEGACO/1 [216.33.33.61]: 27000

```

Transaction = 1235 {
  Context = - {
    Modify = TermA {
      Signals {cg/dt},
      DigitMap= Dmap1 {(2XXX)},
      Events = 1112 {
        al/on, dd/ce {DigitMap=Dmap1}
      }
    }
  }
}

```

MEGACO/1 [209.110.59.34]: 25000

```

Reply = 1235 {
  Context = - {Modify = TermA}
}

```

MEGACO/1 [209.110.59.34]: 25000

```

Transaction = 2001 {
  Context = - {
    Notify = TermA {
      ObservedEvents =1112 {
        20010202T10010000:dd/ce {ds="2992", Meth=FM}
      }
    }
  }
}

```

MEGACO/1 [216.33.33.61]: 27000

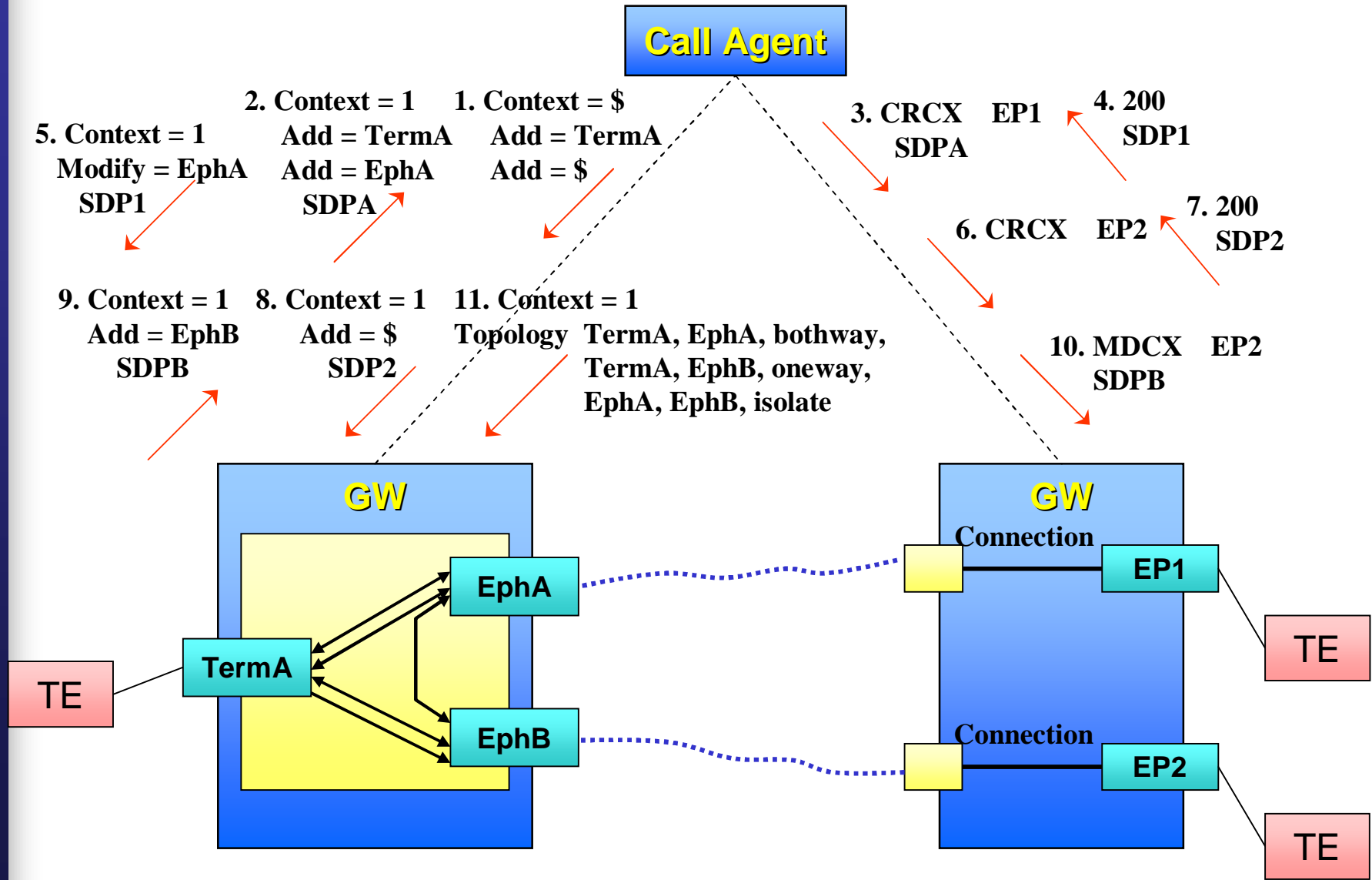
```

Reply = 2001 {
  Context = - {Notify = TermA}
}

```



# Megaco协议基本操作模型





## Megaco消息处理总结

- Megaco/H.248从一个更抽象的角度对网关的控制方式进行了定义（**Termination**、**Context**）
- 以“**Termination**”为核心的网关管理
  - 以Signal和Event为基本控制元素
- 以“**Context**”为核心的Termination管理
  - Context的Topology简化多方管理
  - Move方法简化Termination在Context间的移动
- 以“**Transaction**”为基本消息处理单位
- 以Package完成功能组织

# MGCP与Megaco的差异



- 控制模型的差异
  - Endpoint & Connection
  - Termination & Context
- 控制能力的差异
  - 针对单个端点的控制
  - 对单个端点的控制&对端点间拓扑结构的控制



北京邮电大学  
BEIJING UNIVERSITY OF  
POSTS AND  
TELECOMMUNICATIONS

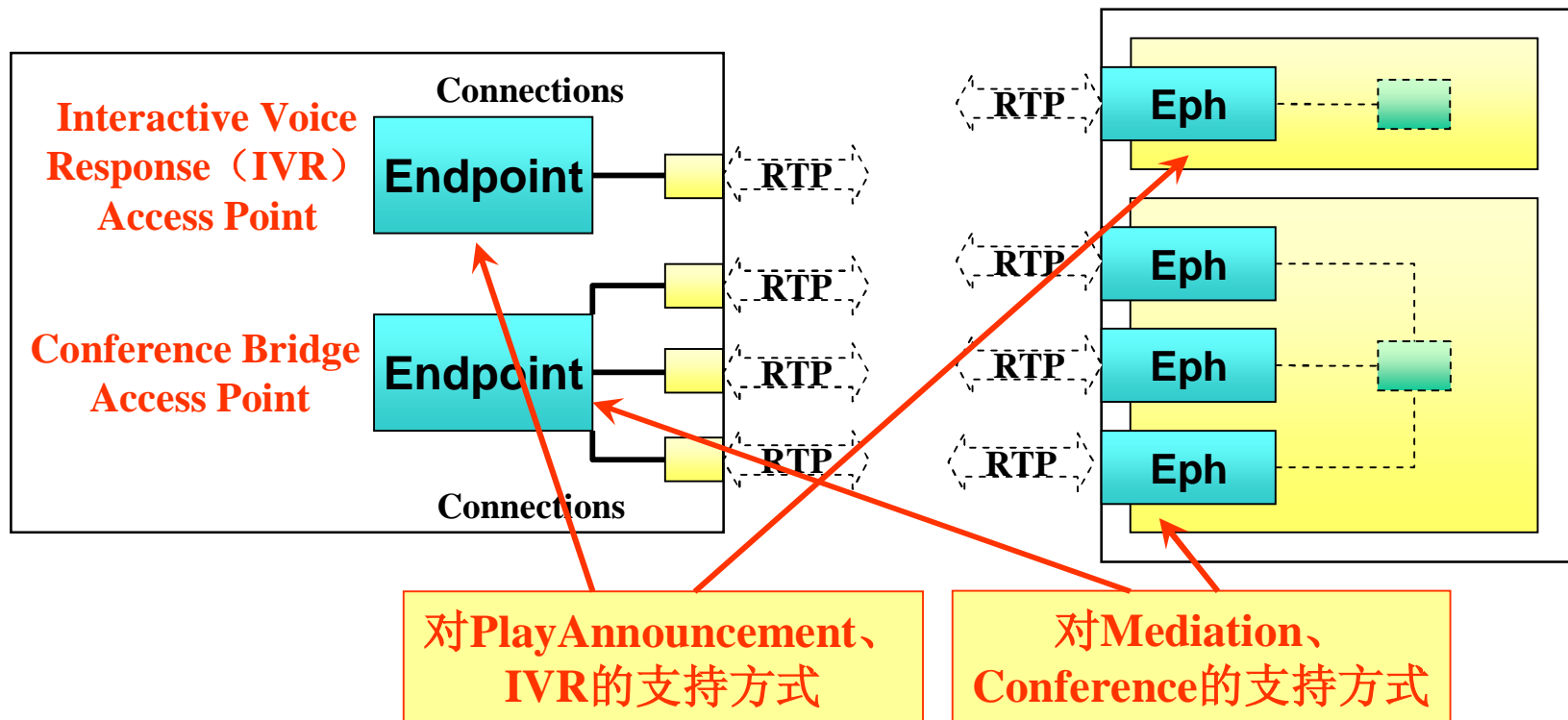
STATE KEY LABORATORY OF SWITCHING TECHNOLOGY AND  
TELECOMMUNICATION NETWORK



# MGCP / Megaco 协议 分析总结 - MS 控制模型



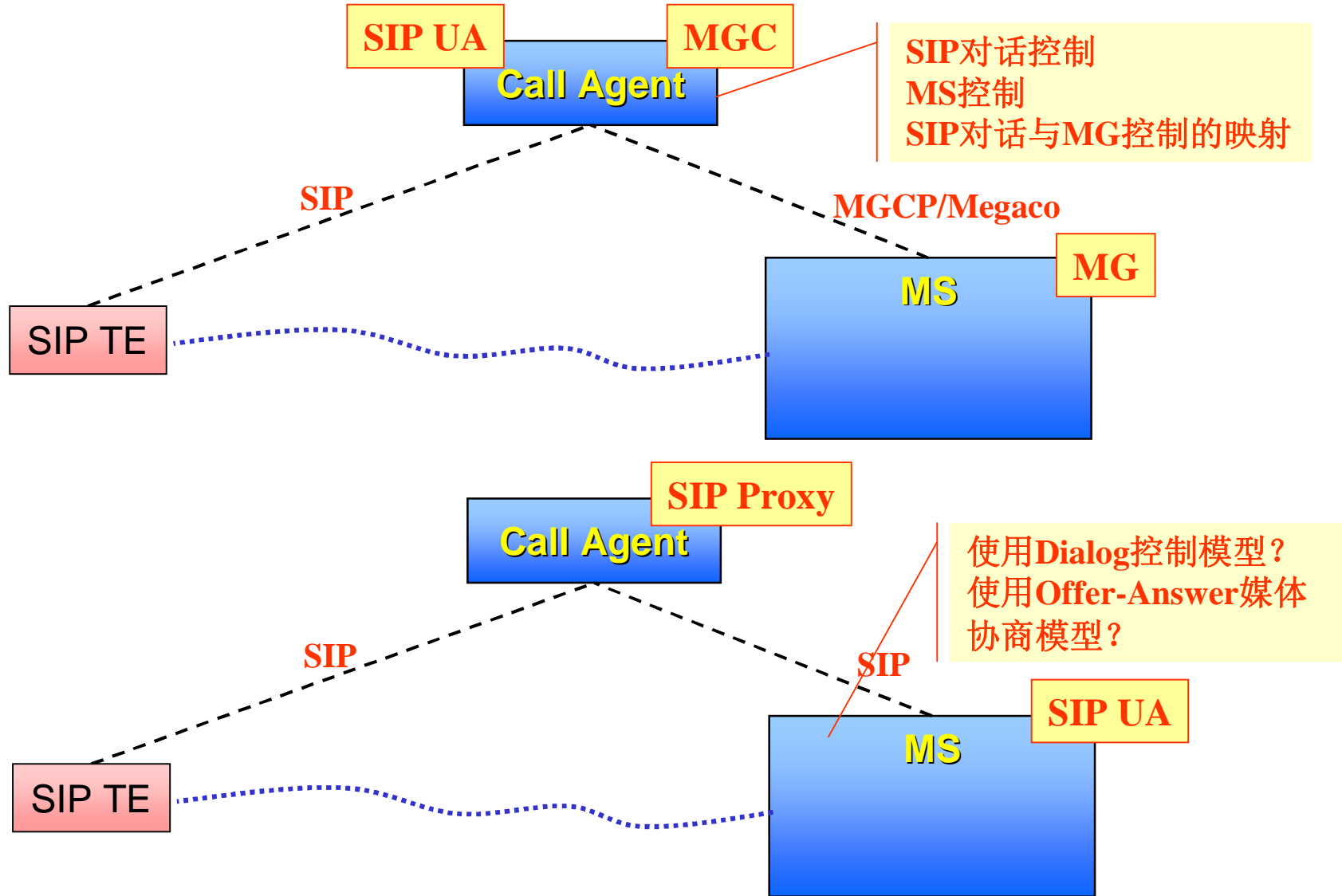
# MGCP/Megaco对媒体服务器的支持



控制能力	MGCP	Megaco	控制能力	MGCP	Megaco
承载连接的媒体协商	MGC控制	MGC控制	多方服务的连接建立	MGC针对多方服务端点的连接建立	MGC针对Context的终结点加入
服务动作的控制	MGC针对服务提供端点的控制	MGC针对连接终结点的控制	多方服务拓扑结构控制	MGC针对多方端点每一个连接属性控制	MGC针对Context拓扑结构控制

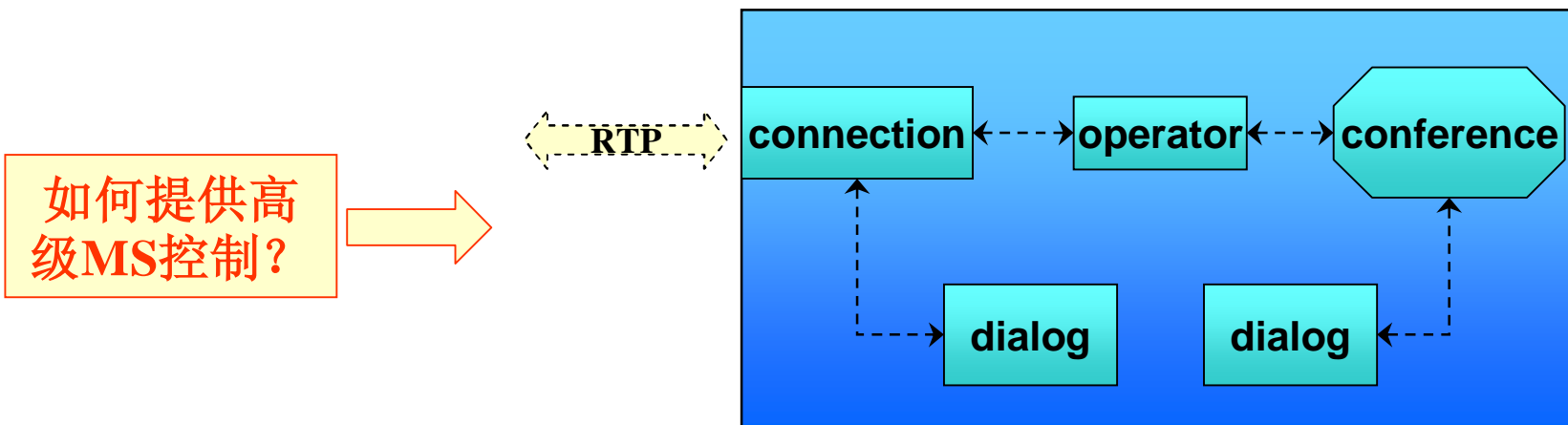


# 使用SIP协议作为MS控制协议？





# 一种新的MS控制模型



- 媒体服务器控制能力:
- 1、承载连接的媒体协商
  - 2、服务动作的控制
  - 3、多方服务的连接建立
  - 4、多方服务拓扑结构控制

都通过扩展SIP协议来完成吗?

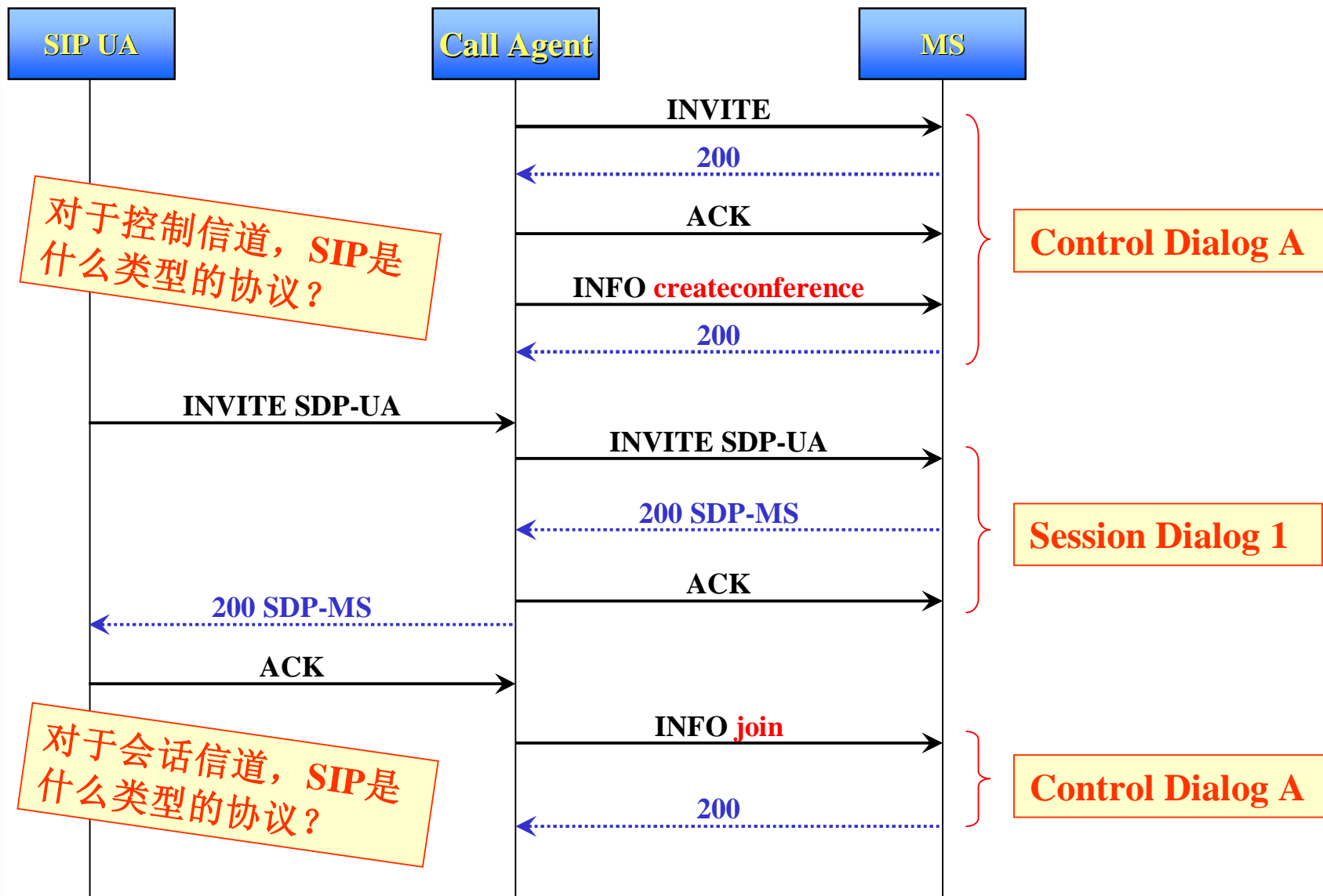
- MSML/MOML
- MSCL
- Media Objects Markup Language
- Media Sessions Markup Language
- Media Server Control Markup Language

SIP做什么?

- 网络连接的建立
- 控制连接的建立



# 以SIP协议为基础的MS控制过程





## 如何看待一个协议？

- 从整体角度看问题
- 系统的体系结构模型？
- 协议在体系结构中所处的位置？
- 协议在体系结构中所起的作用？