

ISAMV协议

笔记本： ISAMV工作记录
创建时间： 2015/12/15 17:01

更新时间： 2015/12/21 16:56

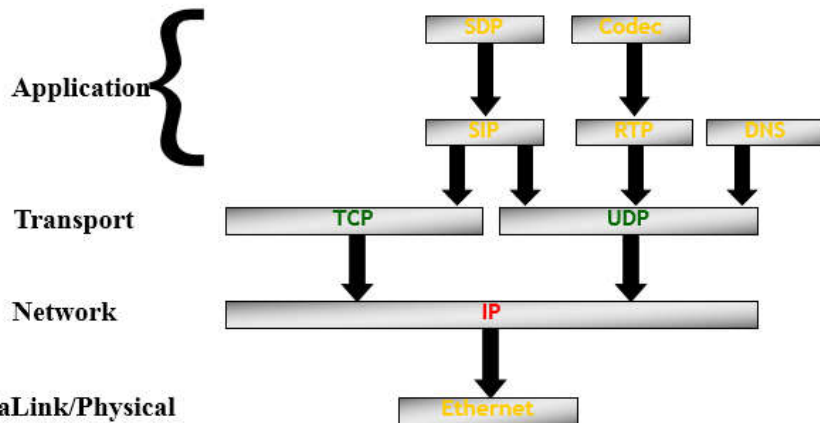
ISAMV协议

2015年12月15日
17:01

• 1, 协议自学

• 1.1 RTP协议

- VOIP的OSI网络模型:



DataLink/Physical

• VoIP = Signaling (SIP+ISUP..) +Media (SDP+RTP+....)

- RTP: 实时传输协议, 端到端的实时传输服务。主要使用UDP协议, 不保证传输的正确和传输的顺序, 有时也会用到TCP协议。
- 时间戳= (当前时间-起始时间) * 采样频率;
- RTP传输的基本过程:
- Payload:
- 模拟语音信号——(通过DSP处理器)——》PCM数字信号流, 数字信号流会通过codec进行适当的压缩——》通过codec, 装配程序或取出信号流中的帧创建数据包, 然后添加RTP的头部(通常包含时间戳, 序列号等)——》添加UDP头部和IP头部(主要包括源地址和目标地址)

IP头部	UDP头部	RTP头部	数据帧
------	-------	-------	-----

- Payout:
- 微处理器去除数据包的IP和UDP头部——》DSP处理器去除RTP头部——》得到语音帧
- RTP协议存在的问题:
 - 会出现网络延迟, 包到达的时间跨度会不确定, 通过DSP jitter buffer去除延迟
 - 包丢失: 一般通过前一个成功接收的包填充丢失的包
 - 顺序错误: 和包丢失相关, 通过前一个成功的包填充位置, 后来到达的包丢弃
- Echo: 指自己听到自己的声音, 主要出现在4线变2线介质发生变化时, 一般使用MGW来消除VOIP中的echo现象。
- VAD和CNG可以压制噪音, 不理睬用户静默阶段的数据, 可以节省带宽, 也使延迟听起来更加自然。

• 1.2 IMS

- IMS: 下一代网络架构的核心, 意为IP多媒体子系统, 基于SIP、SDP、Diameter、H.248等协议。IMS融合了用户数据库(HSS)、业务平台、接入、QoS、承载, 是网络融合FMC的唯一武器, 是全业务的基础。目前各组织统一的是common ims。

- IMS用到的主要协议：
 - SIP: 首选的会话控制协议，基于C/S模式，多媒体应用层控制信令
 - Diameter: HS和会话控制之间交换用户信息
 - H.248: 控制MGW, MGC和MRFP之间信令通信
- IMS主要功能实体：

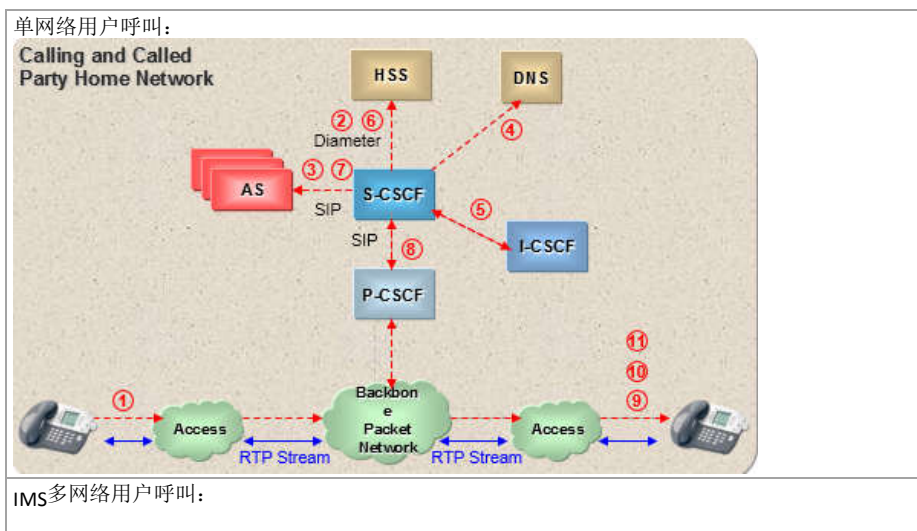
●IMS核心功能实体： <ul style="list-style-type: none"> 呼叫会话控制功能 (CSCF) 归属用户服务器 (HSS) 多媒体资源功能控制器 (MRFC) 多媒体资源功能处理器 (MRFP) 签约定位器功能 (SLF) 策略决策功能 (PDF) 计费相关的功能实体 (CGF) 	●IMS互通功能实体： <ul style="list-style-type: none"> 信令网关 (SGW) 媒体网关控制功能 (MGCF) IP多媒体-媒体网关功能 (IM-MGW) 出口网关控制功能 (BGCF)
●IMS应用功能实体： <ul style="list-style-type: none"> 应用服务器 (AS) <ul style="list-style-type: none"> MM AS, XDM, Presence, IM, PoC, IP Centrex, Video Conf, VCC 智能业务出发服务器 (IM-SSF) 业务应用网关 (OSA-SCS) 	●IMS辅助功能实体： <ul style="list-style-type: none"> ENUM服务器 DNS服务器 网管服务器

多个逻辑功能实体可以集成在公共的硬件平台上

P-CSCF功能： <ul style="list-style-type: none"> IMS中UE的接入点 SIP proxy 和PDF结合完成承载QoS的控制 计费关联，生成CDRs
I-CSCF功能： <ul style="list-style-type: none"> IMS中外网的入口（功能在弱化，可能被IBCF代替） 在UE注册时分配S-CSCF 为来电呼叫决定对应的S-CSCF 产生计费CDRs
S-CSCF功能： <ul style="list-style-type: none"> 通过和HSS交互完成对UE的鉴权和注销 业务控制 路由：对各种网络下呼叫的路由

BGCF: 用于传统的PSTN网络的出口网关控制功能
MGCF: 控制通话状态，使用SIP和I-CSCF/BGCF通信，ISUP和SIP之间的协议转换

- IMS呼叫流程：



• 1.3 SIP协议

• 1, SIP组成实体

- SIP: 初始会话协议, 用于创建、修改和终止会话, 只起到会话保持的作用, 会话本身不需要SIP完成。SIP要求在TCP和UDP下应用, 有些时候鉴于安全会使用TLS和SCTP。
- SIP实体组成:

客户端	用户客户端 (UAC)
服务器	用户服务器(UAS): SIP处理的参与者 (从请求道获得最终回复的过程) 代理服务器: Stateless、Statefull和Call Stateful 重定向服务器: 推送路由信息到客户端 注册器
Stateless代理服务器并不包含一个事务层, 他们直接从通讯录接收请求和发送应答。不允许发送临时应答, 必须忽略ack请求和cancel请求。	

• 2, 请求和响应

- SIP请求: 除了ACK, 其他命令都需要一个最终响应

INVITE	请求一个会话	ACK	确定invite请求的最终响应
Option	选择/查询服务器的能力	Bye	终结会话
Cancel	取消, 用于最终回复前	Register	注册

- 除了基本的6个命令之外, SIP扩展提供了其他的SIP命令, 比如:

PRACK	临时 ACK, 一般用于临时响应的确认	RFC 3262
UPDATE	Update SDP.	RFC 3311
REFER	Refer requestor to another URL.	RFC 3515
SUBSCRIBE	Subscribe for event notification.	RFC 2848
NOTIFY	Event notification.	RFC 3265
MESSAGE	Instant Messaging.	RFC 3428

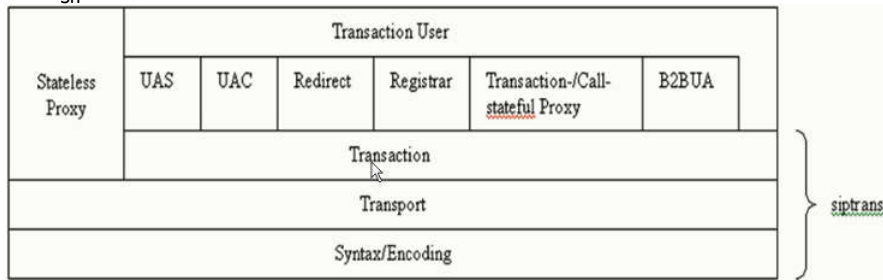
- SIP响应:

1xx	正在处理	2xx	处理成功
3xx	需要转发到其他服务器	4xx	请求中有错误
5xx	服务器错误	6xx	全球性错误

- **1xx Informational**
 - 100 Trying
 - 180 Ringing
 - 181 Call Is Being Forwarded
 - 182 Queued
 - 183 Progress
 - **2xx Success**
 - 200 OK
 - 202 Accepted
 - **3xx Redirection**
 - 300 Multiple Choices
 - 301 Moved Permanently
 - 302 Moved Temporarily
 - 305 Use Proxy
 - 380 Alternative Service
 - **4xx 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
 - 415 Unsupported Media Type
 - 416 Unsupported URI Scheme
 - 421 Extension Required
 - 423 Interval Too Brief
 - 480 Temporarily Unavailable
 - 481 Call/Transaction Does Not Exist
 - 482 Loop Detected
 - 483 Too Many Hops
 - 484 Address Incomplete
 - 485 Ambiguous
 - **5xx Server Error**
 - 486 Busy Here
 - 487 Request Terminated
 - 488 Not Acceptable Here
 - 491 Request Pending
 - 493 Undecipherable
 - 500 Server Error
 - 501 Not Implemented
 - 502 Bad Gateway
 - 503 Service Unavailable
 - 504 Server Time-out
 - 505 Version Not Supported
 - 513 Message Too Large
 - **6xx Global Error**
 - 600 Busy Everywhere
 - 603 Decline
 - 604 Does Not Exist Anywhere
 - 606 Not Acceptable
- Alcatel-Lucent

3. SIP协议格式

- SIP协议格式:

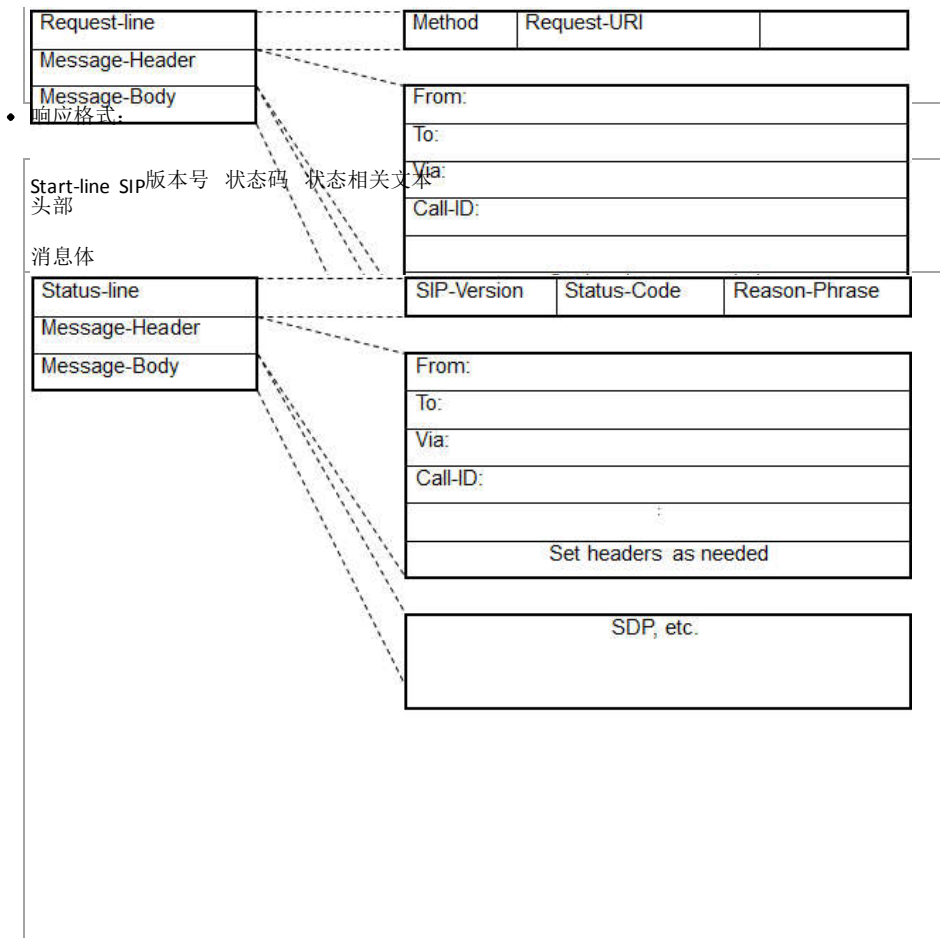


- SIP消息格式:

- 请求格式:

- 一个合法的消息请求至少包含如下头域参数: To, from, cseq, call-ID, max-forwards, via。这6个参数提供了用于路由用的核心信息。
- **Via:** 通知其他实体响应发送到这里, via可以有多个。在请求路由转发的过程中, 每个SIP实体都来设置, 将其地址写入via。Via: 发送协议 发送方; 隐藏参数; 生存期参数; 多播地址参数; 接收方标记, 分支参数
- **Contact:** 通知其他实体将来请求直接发送到这里。Contact: 地址; q参数; 动作参数; 失效参数; 扩展属性
- 在非invite请求下, UAS一般不发送临时应答给请求者
- 在一个显存的对话中发出invite请求是re-invite, re-invite请求可以同时更改对话和会话参数

Start-line	请求方法	request-URI	SIP版本号
头部			
消息体			
其中request-URI的格式为:			
SIP: 用户名: 口令@主机: 端口; 传送参数; 用户参数; 方法参数; 生存期参数; 服务器地址参数? 头部名=头部值			



- TAG: To和from域中在对话建立之后会包含一个TAG标签。tag称为标记，为16进制数字串，中间可带连字符“-”。当两个共享同一SIP地址的用户实例用相同的Call-ID发起呼叫邀请时，就需用此标记予以区分。标记值必须全局唯一。To tag, from tag和Call-id一起完整定义了通话两端之间的关系，并且表示这是一个对话性质的关系。
- Branch: branch是包含在via请求中，用于区分请求创建的事务，客户端和服务端都会使用，在时间和空间上是唯一的。相当于事务的ID。
- SDP: 用于描述SIP消息体的格式，SDP是一种描述格式，可以用于SIP,SAP,MGCP,EMAIL,HTTP等协议，使用下面的格式（下面列出的是必须的参数）：
 - o=用户名 对话id 对话版本 网络类型 网址类型 网址 ——对话描述
 - t=0 0 ——时间描述
 - m=多媒体类型 端口号 传输类型 格式 ——媒体描述
 - a=rtpmap: 编码类型 编码名/采样频率/编码参数
- SDP通过问答模式来确定对话双方使用的codec和船体的媒体类型，一般SDP offer包含在invite命令中，也有其他情况会包含SDP offer:

Offer	Answer	RFC
1. INVITE Req.	2xx INVITE Resp.	RFC 3261
2. INVITE Req.	1xx-rel INVITE Resp.	RFC 3262
3. 1xx-rel INVITE Resp.	PRACK Req.	RFC 3262
4. PRACK Req.	200 PRACK Resp.	RFC 3262
5. 2xx INVITE Resp.	ACK Req.	RFC 3261
6. UPDATE Req.	2xx UPDATE Resp.	RFC 3311

- SIP头域中有3个参数用于指定消息体的特征：
 - ContentType: 消息体描述类型，一般是application/sdp
 - ContentLength: 消息体的长度，byte为单位
 - contentEncoding: 压缩时使用的编码类型
- SPI-T:

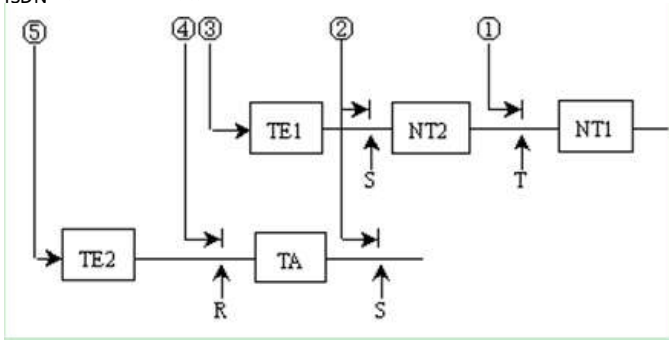
- SIP-T并不是一个新的协议，它在SIP的基础上增加了关于如何实现SIP网络与PSTN网络互通的扩展机制，包括三种应用模型：PSTN-IP、IP-PSTN、PSTN-IP-PSTN。
- SIP-T协议的特点：
 - 封装：在SIP消息体中携带ISUP消息；
 - 映射：ISUP-SIP消息映射，ISUP参数与SIP头域映射。
- SIP消息与ISUP信令之间的映射关系可简单描述为：
 - IAM = INVITE
 - ACM = 180 RINGING
 - ANM = 200 OK
 - RLS = BYE
 - RLC = 200 OK

1.4 ISDN协议

1， ISDN

- ISDN被称为综合业务数据网，利用单一的通信网络实现包括语音。文字。数据。图像等在内的各种业务；提供了端到端的数字连接，为用户入网提供多用途的网络接口。功能群指用户接入ISDN所需的一组功能；参考点是用来分割功能全的概念上的点。
- ISDN的信道分为两种：B信道：64kbps，用户信息传送信道，电路交换/分组交换/专线；D信道：16或64kbps，用于电路交换信令。目前用户接入网络的接口类型有以下两种：
 - BRI：基本速率接口，2B+D，192kbps，
 - PRI：基群速率接口，30B+D，2048kbps

- ISDN用户系统的结构如下：



功能群		功能
NT1	Network termination 1 (网络终端设置1)	用户终端；第一层功能维护和监测；时钟同步；供电；第一层多路复用；接口终端
NT2	Network termination 1 (网络终端设置1)	第二、三层协议处理；第二、三层复用；交换；集线；维护功能；接口终端
TE	Terminal equipment	协议处理
TE1	Terminal equipment type1 (ISDN标准终端)	维护功能； 接口功能
TE2	Terminal equipment type2 (ISDN非标准终端)	和其它设备的连接功能
TA	Terminal adaptor (终端适配器)	接口转换功能，是现有终端功能过渡到DSS1的桥梁

- ISDN用户网络接口协议的结构如下：

网络层	I. 451(Q. 931)	X. 25分组级	IP
链路层	LAPD(Q. 921)		PPP
物理层	I. 430(BRI)或I. 431(PRI)		
	电路交换	分组交换	B信道
	D信道		

-
- [RB1]
- ISDN有以下三种不同的信令：
 - 用户—网络信令：用户终端和网络之间的控制信号
 - 网络内部信令：交换机之间的控制信号
 - 用户—用户信令uus：在用户终端设备之间的控制信号，网络不做任何处理
-

• 2，Q.931协议

- Q.931协议描述的是D通道第三层的各种功能，说明了在ISDN用户_网络接口处的呼叫建立、进行和拆线的过程。931需要数据链路层Q.921提供的功能和服务来实现功能。
- Q931的主要功能：
 - 处理与数据链路层通信的原语
 - 产生和解释同层通信的第三层消息
 - 管理呼叫控制过程中使用的定时器和逻辑实体
 - 接入资源的管理，包括B通道的管理
 - 保证所提供的业务与用户要求一致性的检查，比如承载能力
 - 选路和中继，传递用户和网络间的信息

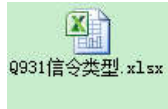
• 1，消息类型

- Q.931协议的消息在不同的mode下可以使用的消息并不相同，有的消息在某些mode下不被支持。Q931主要有4种模式：circuit-mode、packet-mode、user signalling bearer service(temporary signalling connection)、global call reference。[BR2]
- 普通电路交换通话中的消息类型：

1，呼叫建立消息	
Alerting	提醒，表示用户开始振铃 Direction: Called user -> network -> calling user
Call proceeding	呼叫进程，表示所请求的呼叫已经建立，并不再接受呼叫建立消息 Direction: Called user -> network -> calling user
Connect	连接，表示被叫用户接受呼叫 Direction: Called user -> network -> calling user
Connect ack	连接证实，表示该用户得到呼叫 Direction: Calling user -> network -> called user
Progress	进展，用户或网络发送，表示呼叫进展
Setup	建立，主叫发送，启动接入连接的建立 Direction: Calling user -> network -> called user
Setup ack	建立证实，表示呼叫建立开始 Direction: Called user -> network -> calling user
2，呼叫信息阶段消息	
Suspend	暂停，用户请求网络暂停呼叫 Direction: User -> network
Suspend ack	暂停证实，完成暂停呼叫 Direction: Network -> user
Suspend reject	暂停拒绝，网络发送，暂停呼叫不成功 Direction: User -> network
Resume	恢复，用户请求恢复暂停的呼叫。同理有ack和reject Direction: User -> network
3，呼叫消除消息	
Disconnect	拆线，用户发送，请求网络断开连接，或网络发送
Release	用户或网络发送，准备释放通路和呼叫参考
Release complete	用户或网络发送，资源释放完毕
4，其他消息	

Information	信息，用户或网络发送，提供附加信息，比如电话号码等
Notify	通知，用户或网络发送，通知与呼叫相关的信息
Status	网络发送，响应状态查询 Direction: Network -> user
Status enquiry	状态询问，请求网络的状态信息，必须用status响应 Direction: User -> network

- 其他模式下的消息类型：



• 2，消息结构

- Q.931消息组成结构：

协议鉴别语 [RB3]		
0 0 0 0		呼叫参考值的长度 (八比特组)
呼叫参考值		
0	消息类型 [BR4]	
所需要的其它信息单元 [BR5]		

- Call reference: 呼叫参考，区别用户网络接口上的呼叫，标识B信道上的一个呼叫，在整个呼叫期间参考值不变，呼叫结束或通话暂停时释放。呼叫参考值单元的最大长度可以达到2个octet（不包括length），结构如下：

8	7	6	5	4	3	2	1	Octet
0	0	0	0	Length of call reference value (in octets)				1
Flag		Call reference value						2
								etc.

[BR6]

- Information elements: 信息单元分为两类，一类是单个octet的单元，一类是长度可变的单元（占据多个octet）。起格式如下：

8	7	6	5	4	3	2	1	Octet
1	Information element identifier			Contents of information element				1

a) Single octet information element format (type 1)

8	7	6	5	4	3	2	1	Octet
1	Information element identifier							1

b) Single octet information element format (type 2)

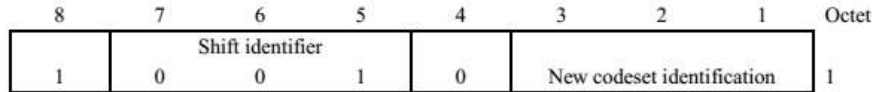
8	7	6	5	4	3	2	1	Octet
0	Information element identifier							1
Length of contents of information element (octets)								2
Contents of information element								3 etc.

c) Variable length information element format

[BR7]

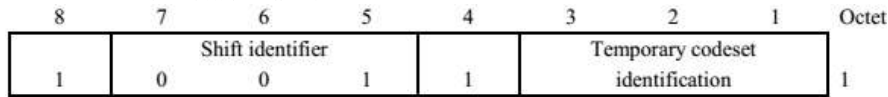
- Locking shift: 锁定转换，用更高的编码集合来说明接下来的信息单元，直到下一个锁定转换出现

时失效，初始编码集为集合₀。格式如下：



"0" in this position
indicates locking shift

- [BR8]
 - Non-locking shift: 非锁定转换，编码集合转化尽用来说明下一个信息单元，之后的灰继续采用现行的集合来解释。格式如下：



"1" in this position indicates
non-locking shift

- Q931协议的示例：

以不同的颜色标识同一个消息内的不同内容，以相同的颜色标识相应的八位字节编码。

Protocol-discriminator call-reference message-type bearer-capability channel-identification
calling party number called party number

传输格式：

```
< 1: ( 42) 2751332
< 00 01 b8 54 46 02 00 1a 05 04 03 80 90 a3 18 02
< ca 82 6c 0c 00 80 31 30 30 31 30 30 30 30 30
< 70 08 80 38 38 30 30 30 30 30 30
```

解析后格式：

```
LAPD: 0 R (F) I-frame (NR=2a NS=5c)
Q.931-BRI: ref=0x001a SETUP
Bearer cap[BR9] (3)
Channel Id[BR10] (2)
Calling Party Number[BR11] (12) 1001000000
Called Party Number [BR12] (8) 8880002
```

- ISDN basic call 呼叫的流程及其931消息：



- 3, 主叫流程

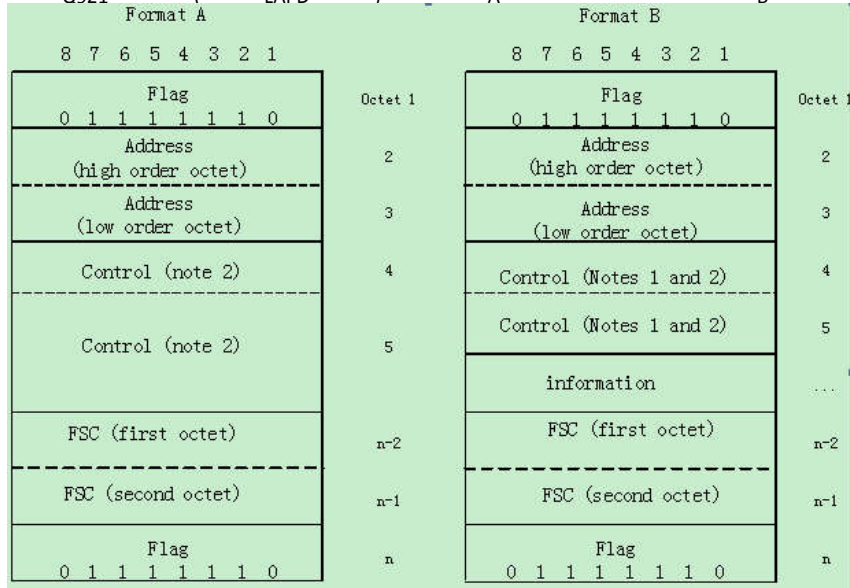
3, Q.921协议

- LAPD(Link Access Protocol on D Channel)是D信道的数据链路层协议，包含在Q921中。LAPD的目的是通过用户—网络接口在第三层实体间传递信息，LAPD的主要功能有：
 - 帧的分割、同步和透明传送。
 - 同一D通路上多个数据链路的复用。
 - 保持帧接收的顺序和发送的顺序一致。
 - 检测数据链路上传输错误，格式错误和操作错误。
 - 用重发的方法来纠正传输错误。
 - 当发生了不可纠正的错误时通知管理实体。
 - 进行流量控制。
 - 物理层的激活、去激活管理

- Q921说明的是LAPD帧结构、规程单元和字段格式。Q921的协商过程可以分为分配TEI、建立多帧操作（建链），维护链路、拆除链路四步。

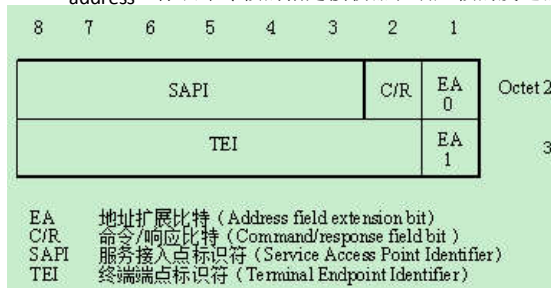
1, 帧结构

- Q921的帧结构(也就是LAPD的结构)如下, 格式A用于无信息字段的帧, 格式B用于有信息字段的帧:



2, 规程单元和字段格式

- Flag: 所有帧都是以01111110开始和结尾的
- address: 标识命令帧的指定接收器和响应帧的发送器, 标志D信道上多个数字链路。格式如下:



[BR13]

- control: 用于识别帧的类型, 标记是命令还是响应。Control格式分为编号信息传送(I帧)、监视管理(S帧)、无编号信息传送和控制(U帧)。

- I帧——用于第三层实体之间传递消息;
- S帧——用于执行数据链路的监视和管理功能, 比如确认I帧, 请求重发I帧等
- U帧——用于提供附加数据链路控制功能和无确认信息传送的无编号信息传送

控制字段比特	8	7	6	5	4	3	2	1	
I格式	N(S)							0	Octet 4
	N(R)							P	5
S格式	X	X	X	X	S	S	0	1	Octet 4
	N(R)							P/F	5
U格式	M	M	M	P/F	M	M	1	1	Octet 4
N(S) 发送I帧的发送序号									M 修改功能比特
N(R) 下一个期望接受的I帧的发送序号									X 保留, 设置为0
S 监视功能比特									
P/F 发送的帧为命令帧时为探测比特; 发送的帧为响应帧时为终止比特									

[BR14]

- information: 仅在I和某些U帧中出现, 包含的是用户的数据
- FSC: 帧校验序列, 检查是否有错误的错误校验码

Q921字段映射规则:

- 对于单个8位bit组包含的字段, 最小的bit代表最小的顺序值[BR15]
- 对于跨越两个bit组的字段, 最高顺序的bit在起始的bit位置, 最低的映射在结束的bit

- 对于完全占据两个bit组的字段，第一个bit组的bit 1最大，第二个bit组的bit 8最小[BR16]。[BR17]

3，帧类型

- Q921命令帧和响应帧的类型：

Application	Format	Commands	Responses	Encoding								Octet	
				8	7	6	5	4	3	2	1		
Unacknowledged and Multiple Frame Acknowledged Information Transfer	Information transfer	I (information)		N(S)							0	4	
				N(R)							P	5	
	Supervisory	RR (receive ready)	RR (receive ready)	0	0	0	0	0	0	0	1	4	
				N(R)							P/F	5	
		RNR (receive not ready)	RNR (receive not ready)	0	0	0	0	0	0	1	0	1	4
				N(R)							P/F	5	
		REJ (reject)	REJ (reject)	0	0	0	0	0	1	0	0	1	4
				N(R)							P/F	5	
	Unnumbered	SABME (set asynchronous balanced mode extended)											
				0	1	1	P	1	1	1	1		4
			DM (disconnected mode)										
				0	0	0	F	1	1	1	1		4
		UI (unnumbered information)		0	0	0	P	0	0	1	1		4
				0	1	0	P	0	0	1	1		4
			UA (unnumbered acknowledgement)	0	1	1	F	0	0	1	1		4
				1	0	0	F	0	1	1	1		4
Connection management		XID (Exchange Identification) (Note)	XID (Exchange Identification) (Note)	1	0	1	P/F	1	1	1	1	4	

- I: information, 通过数据链路连接传送第三层信息的编号帧
- RR: receive ready, 表示数据链路层实体的状态，包括：已准备好接收I帧、确认此前已接收到编号 $\leq N(R)-1$ 的帧、清除先前通过RNR帧表示的忙状态。数据链路层实体还可以用P为“1”的RR命令帧对其对端数据链路层实体的状态
- RNR: receive not ready, 本方处于忙状态，但编号不大于 $N(R)-1$ 的全部帧能得到确认
- REJ: reject, 拒绝帧，请求双方重发从编号 $N(R)$ 开始的I帧
- SABME: set asynchronous balanced mode extended, 将编址的用户侧或网络侧置于模128的多帧确认操作。建立逻辑连接请求
- UI: unnumbered information, 第三层或管理实体请求无确认信息传送时，无编号命令UI将信息发送到对端
- DISC: disconnect, 拆线断开请求
- XID: exchange identification, 工作参数的协商和修改
- UA: unnumbered ack, 响应SABME或DISC命令的接收和接受
- DM: disconnected mode, DM帧来向对端报告其处于不能实现多帧操作状态
- FRMR: frame reject, 报告不能通过重发同一帧来恢复差错状态[BR18]。

4，原语和层间通信

- 在各个层之间是通过原语来进行通信的，原语命令的格式如下：
 - 接口名 - 属性名[BR19] - 类型[BR20]：参数[BR21]
- 其中接口名根据层的不同有所不同：
 - DL: 层3和数据链路层通信
 - PH: 数据链路层和物理层通信
 - MDL: 层管理和数据链路层通信
 - MPH: 层管理和物理层通信
 - 4种类型及其含义：

4.1.2.1 request

The request primitive type is used when a higher layer or layer management is requesting a service from the lower layer.

4.1.2.2 indication

The indication primitive type is used by a layer providing a service to inform the higher layer or layer management.

4.1.2.3 response

The response primitive type is used by layer management as a consequence of the indication primitive type.

4.1.2.4 confirm

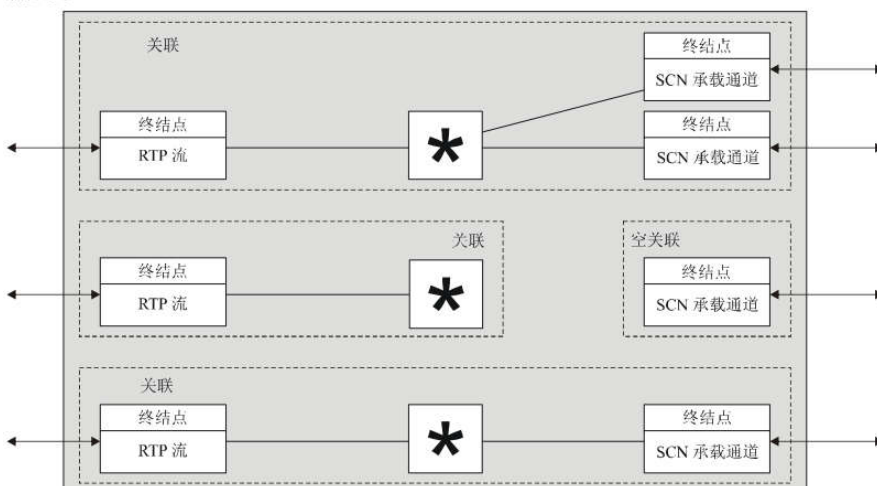
The confirm primitive type is used by the layer providing the requested service to confirm that the activity has been completed.

- 5. P2P

• 1.5 H.248协议(非工作范围)

- MG: 媒体网关, 可以将一种网络类型的媒体转换为理工一种网络类型所要求的媒体
- Trunking gateway: 在电路交换网络和分组网络之间中介大量数字电路的网关
- PSTN: 公众交换电路网; SCN: 电路交换网络
- 连接模型:
- 可以被MGC控制的逻辑实体对象, 由终结点和关联组成。终结点用于发起和接收媒体流, 关联是一组终结点间的联系

媒体网关



- 关联的特征参数:

标识符	contextID	紧急呼叫标识符	优先处理
拓扑描述符	某个终结点接受和发送流向	IEPS呼叫标识符	业务和技术的细节
优先级	0最小, 1最大	关联特征描述符	扩展特性

- 终结点:

- 终结点第一次被创建时, 除了 `terminationstate` 和 `localcontrol`, 其他描述符的属性值应该为空或无参数值。

- 根终结点作用于整个媒体网关, 并且仅能使用如下命令:

令: `modify`, `notify`, `auditvalue`, `auditcapabilities`, `servicechange`。

- 命令:

- 大多数命令是命令网关控制器发起, MG作为命令响应方的; 不过 `notify` 是MG发送给MGC的, `servicechange` 是由MG或MGC发起的

- `add`: 向一个关联添加终结点
- `modify`: 修改终结点的属性, 事件, 信号, 不能使用 `choose` 通配符
- `subtract`: 删除关联中的终结点, 并返回当前统计值, 不能使用 `choose`
- `move`: 将终结点转移到另一个关联, 不能操作空关联
- `auditvalue`: 获取终结点当前属性, 事件, 信号, 统计值
- `auditcapabilities`: 获取MG所允许的终结点的属性, 事件, 信号所有可能值
- `notify`: 将检测到的时间上报MGC
- `servicechange`: MG——》MGC: 终结点退出服务, 终结点返回服务, MG可用

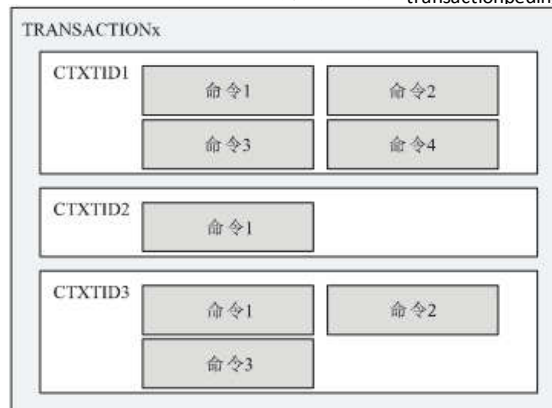
性：MGC——》MG：开始/完成启动，MGC切换，使终结点退出/返回服务

- 描述符：
- 作为命令的输入或输出参数，由描述符名+参数组成，格式如下：
- 描述符名=<someID>{param=value,param=value,...}
- 描述符有以下常用：

描述符名称	说 明
调制解调 (Modem)	标识所使用的调制解调器类型和属性 (注)
复用 (Mux)	描述多媒体终结点的复用类型(如H.221、H.223和H.225.0)和组成复用终结点的终结点
媒体 (Media)	媒体流的列表 (参见7.1.4)
终结点状态 (TerminationState)	包中所定义的与特定媒体流无关的终结点属性
流 (Stream)	单个媒体流的远端/本地/本地控制描述符的列表
本地 (Local)	包含媒体网关从远端实体接收到的媒体流属性
远端 (Remote)	包含媒体网关发送至远端实体的媒体流属性
本地控制 (LocalControl)	包中所定义的包含与媒体网关和媒体网关控制器有关的一些属性。
事件 (Events)	描述可以被媒体网关检测的事件，以及当事件被检测发生时的处理机制。
事件缓存 (EventBuffer)	描述当事件缓存处于激活状态时，可以被媒体网关所检测的事件
信号 (Signals)	描述适用于终结点的信号 (参见7.1.11)
审计 (Audit)	适用于Audit命令，描述需要审计的信息
包 (Packages)	适用于AuditValue命令，返回由终结点实现的包的列表
数字映射 (DigitMap)	定义一组特定的事件被匹配的模式，使得事件可以按组而不是单个上报
业务改变 (ServiceChange)	适用于业务改变 (ServiceChange) 命令，描述何种业务发生改变以及业务发生改变的原因
被观察事件 (ObservedEvents)	适用于Notify 或者 AuditValue命令，上报已检测到的事件
统计值 (Statistics)	适用于Subtract、Audit命令，上报与终结点有关的统计数据
拓扑 (Topology)	描述关联中终结点之间的媒体流流向
关联特征 (ContextAttribute)	包中所定义可以影响整个关联的属性
差错 (Error)	包含差错代码和可选的差错描述文本，适用于Notify请求或所有命令响应

- 注一 ITU-T H.248.1建议书第2版 (05/2002) 不再推荐使用 Modem描述符。

- 事务：
- 由一个或多个动作组成，一个动作作用于一个关联的一系列非空命令。构成每个事务应设置一个应用层定时器，对请求消息重传。如果收到transactionpending则重传，收到响应则停止。



• 2，LAB配置

• 0，配置SIP Lab环境

- Server是FS5K时的配置参照：
- //使用默认DB清空之前配置
- admin software-mngt oswp 1 activate with-default-db
- *****
- //默认不需要配置，自动恢复
- configure system management host-ip-address manual:135.251.92.2/25
- configure system management default-route 135.251.92.1
- *****
-
- 0.configure port
- configure mstp general disable-stp
- configure interface shub port [0...7] port-type network admin-status up
-
- 1.configure sig and voice vlan
- configure vlan shub id 18 mode voice-vlan
- configure vlan shub id 18 egress-port network:2[BR22]
- configure vlan shub id 18 egress-port lt:1/1/[1...8]

- configure vlan shub id 18 egress-port lt:1/1/[12...19]
- //configure bridge shub port 0 pvid 18
-
- configure vlan shub id 4093
- configure vlan shub id 4093 egress-port network:2
- configure vlan shub id 4093 untag-port network:2
- configure bridge shub port 2 pvid 4093
-
-
- 2.configure vrf
- configure ip shub vrf 1 fast-path-mode:ena-user-user-com name voice
- configure interface shub vlan-id 18
- configure interface shub ip 18
- configure interface shub ip 18 vrf-id 1
- configure interface shub ip 18 ip-addr 21.1.1.70/24[BR23]
- configure interface shub vlan-id 18 admin-status up
- configure ip shub vrf 1 route-dest 0.0.0.0/0 next-hop 21.1.1.1 vlan-id 18
-
- //=====configure sip application=====
- 1.plan card
- configure equipment slot 1/1/3[BR24] planned-type npot-c capab-profile sip
-
- 2.configure sip server,dial plan and digitmap
- //设置sip server
- configure voice sip server srv_01 admin-domain-name vsp1 port 5060 priority 10 server-role registrar-and-proxy site primary-site address dns:pcgw-stdn.imsgroup0-000.ism01.qd.lucentlab.com[BR25] dnsname-type fqdn
- configure voice sip server srv_01 admin-status up
- configure voice sip dialplan dp_01 pre-activated on
- configure voice sip digitmap dp_01 type regular rule xxxxxxxx
-
-
- 3.configure sip-ua, sip-ua-ap and vsp
- //设置sip UA
- configure voice sip user-agent ua_01 signal-gateway-ip 21.1.1.1 signal-vlan-id 18 admin-status up ip-addr-policy centralized ip-config-mode manual
- configure voice sip user-agent-ap ua_01 slot-id nt signal-ip 21.1.1.70/24
- configure voice sip vsp vsp1 domain-name qd.lucentlab.com[BR26]
- configure voice sip user-agent-ap ua_01 slot-id nt admin-status up
- //设置注册管理器
- configure voice sip register vsp1 register-uri pcgw-stdn.imsgroup0-000.ism01.qd.lucentlab.com
- configure voice sip transport udp provider-name vsp1 admin-status up
- configure voice sip vsp vsp1 admin-status up
- //设置sip的DNS服务器
- configure voice sip dnsserver dns_srv_01 admin-domain-name vsp1 address ipv4:135.252.37.114 [BR27] admin-status down
- configure voice sip dnsserver dns_srv_01 admin-status up
- //设置对话计时器
- configure voice sip system session-timer admin-status up
-
- 4.configure sip user
- //配置sip用户, 包括电话号码等
- configure voice sip termination 1/1/3/1 uri +13288617301@qd.lucentlab.com directory-number 88617301 user-name 13288617301 md5-realm qd.lucentlab.com md5-password passwordlss[BR28] admin-status up
- configure voice sip termination 1/1/3/2 uri +13288617302@qd.lucentlab.com directory-number 88617302 user-name 13288617302 md5-realm qd.lucentlab.com md5-password passwordlss admin-status up
- //=====sip application配置完成=====
-
- 5.configure mirror port:
- configure interface shub mirror enable
- configure interface shub mirror to-port 5
- configure interface shub mirror port lt:1/1/3 internal ingress
- configure interface shub mirror port lt:1/1/3 internal egress
- configure interface shub mirror port lt:1/1/3 external ingress
- configure interface shub mirror port lt:1/1/3 external egress
- //完成
- 使用VT模拟工具的时候需要连接一个小卡, 因此上面的脚本需要进行相应的修改:
 - Singaling ip需要设置成VT中配置的destination的地址
 - Sip server、register-uri、和dns的地址要变成小卡的本地ip地址



- 1, 上传oswp

- Step 1:
到FTP服务器上下载完成的tar文件, 比如: SD_41.055.tar

Step2:
将tar文件解压到本地目录, 目录名可以和tar文件同名, 比如SD_41.055
然后调整结构如下:
目录: SD_41.055
- 目录: L6GQAA41.055
 (该文件放着从原先tar文件解压开的所有文件)
- 文件: L6GPAA41.055
 文件内容: (注意: 135.251.205.41是本机的IP地址)

```
=====
OVERALL-DESCRIPTOR-FILE L6GPAA41.055

BEGIN

SYNTAX-VERSION : 02.00;

ASAM-CORE      : L6GQAA41.055/L6GQAA41.055 135.251.205.41 0.0.0.0;

END
=====
```

Step3:
开启tftp服务器, 将上面的目录作为服务器根目录

Step4:
使用命令:
show software-mngt oswp, 找到一个inactive的配置。如果没有空闲槽位, 需要清空一个槽位, 有两种方式:
a, admin software-mngt oswp 2 abort-download //将槽位二清空
b, admin software-mngt oswp 1 commit //将槽位1提交, then 槽位2 也会被清空

Step5:
下载OSWP配置文件:
configure software-mngt oswp 2 download-set complete-set primary-file-server-id 135.251.205.41
admin software-mngt oswp 2 download L6GPAA41.055
注: 如果下载失败, 使用命令取消下载:
admin software-mngt oswp 2 abort-download

Step7:
下载过程中可以使用show software-mngt oswp查看下载状态
使用新的DB激活OSWP:
admin software-mngt oswp 2 activate with-linked-db
注: 如果选用defaultDB则不使用先前的配置

Step8:
等待重启之后重新配置lab即可

其他命令:
admin software-mngt oswp 2 commit (delete the unuse files?)
info configure software-mngt oswp 2 detail
info configure system security filetransfer detail

如果需要删除OSWP，可以登录到NT上去手工删除文件，然后输入admin software-mngt oswp 2
abort-download命令改成NO_OSWP

• 2，上传CDE

- //-----2,get a new cde and modify the cde-----
- 1. download cde from server to pc to a server and download cde: DIR:/ASB-WBA/views/IMAGE/CDE/R4.3/MSR2GB43.009
 - (RA cde not support NPORT-A, GB support)
 - MSR2GB43.009 0** loose couple 1** tight couple
- 2. download cde from pc to lab on cli
 - admin software-mngt cdeprof download 135.242.96.160:MSR2JC43.001 [local_pc_ip: cde_name]
- now the cde will be download
- 3. on cli, check the download status
 - show software-mngt cdeprof dldstatus
- we can see the path:MSR2GB43.009(new)
- 4. active this cde
 - activate MSR2GB43.009
- 5. we can see the activate statue,check the download status
 - show software-mngt cdeprof dldstatus
- we can see the path:MSR2GB43.009(Activated),now the cde related card will reboot.
- 6. then configure lab and make call
- //-----end get a new cde and modify the cde-----

• 3，修改CDE

//-----3,modify cde and change cde for lab-----

1. unzip this cde. (althouth it is not zip file, we can modify its postfix and unzip)
2. modify the largest 2 file. (MIRDGB43.009, MSS8GB43.009)
3. 如将media server混音改为本地混音 (两个最大的文件都需要改)

```
<Conference>
  <Mode>
    <INVITEWithURIList>
      <IncludeRecipientListInINVITE>Yes</IncludeRecipientListInINVITE>
      <ConferenceFactoryURI>conf-factory</ConferenceFactoryURI>
    </INVITEWithURIList>
  </Mode>
  <ReholdRemotePartBeforeReturnBackToToggler>Yes</ReholdRemotePartBeforeReturnBackToToggler>
```

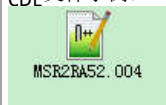
改为

```
<Conference>
  <Mode>
    <INVITEMode/>
  </Mode>
  <ReholdRemotePartBeforeReturnBackToToggler>No</ReholdRemotePartBeforeReturnBackToToggler>
```

4. 将这些内容传到server /home/gavinr/cde目录下
5. 登录到一个sun server 172.24.213.216,create a CDE file
/home/buildmgr/IMAGE/CDE/script/tmptsmalltar.ksh 43.009 /home/gavinr/cde //cde下含四个文件
将在/home/gavinr/下产生一个43.001CDE文件
6. 回传到本地，然后 可以按照get CDE 的步骤进行了

//-----end modify cde and change cde for lab-----

CDE文件示例:



• 4，上传下载DB

- <http://aww.dsl.alcatel.be/ftp/pub/outgoing/ESAM/DAILY/> 下载image和OSWP files 两个版本号码一样

-
- 1.startup the system with the old version.
-
- 2.upload the DB to outside tftp server:
-
- isadmin># admin software-mngt database upload actual-active:135.251.204.206:dm_complete.tar
-
-
- 3.download DB from tftp server.
- admin software-mngt database download 135.251.204.148:dm_complete.tar
-
- 7.active the new OSWP with new DB.
- admin software-mngt oswp 2 activate with-linked-db
-
- 8.delete the unuse files.
- admin software-mngt oswp 2 commit
-
- 9.After reset, check the DB wether it takes effect or not.
- //////////////////////////////////////
- info configure software-mngt oswp 2 detail
- info configure system security filetransfer detail
-
-

• 5，配置mirror

- mirror配置:
-
- 1. 配置mirror口 (NTIO板上的端口)
- configure interface shub mirror to-port 5 enable
- 2. 配置需要映射到该mirror口的端口
- configure interface shub mirror port network:0 external ingress egress
- configure interface shub mirror port network:0 internal ingress egress
- 其中external表示从端口0到外网方向上的包, internal表示从端口0通过lanx到系统内部方向上的包.
- ingress表示该方向上进来的包, egress表示该方向上出去的包.
-
- 3. 也可以配置LT板映射到mirror口
- configure interface shub mirror port lt:1/1/12 external ingress egress
- configure interface shub mirror port lt:1/1/12 internal ingress egress
- configure interface shub mirror port lt:1/1/13 external ingress egress
- configure interface shub mirror port lt:1/1/13 internal ingress egress
- configure interface shub mirror port lt:1/1/19 external ingress egress
- configure interface shub mirror port lt:1/1/19 internal ingress egress
-
-
- 4. 标准抓包配置:
- configure interface shub mirror port network:0 external ingress
-
- configure interface shub mirror port lt:1/1/12 external ingress
- configure interface shub mirror port lt:1/1/12 internal ingress
-
- configure interface shub mirror port lt:1/1/13 external ingress
- configure interface shub mirror port lt:1/1/13 internal ingress
-
- configure interface shub mirror port lt:1/1/19 external ingress
- configure interface shub mirror port lt:1/1/19 internal ingress
-
- NANT-D MIRROR:
- configure port ntio-1:sfp:5 no shutdown
- configure mirror mirror-dest 200 create
- sap ntio-1:sfp:5:870 create
- configure mirror mirror-dest 200 no shutdown
- debug mirror-source 200 port lt:1/1/6 ingress egress
- debug mirror-source 200 port ntio-1:sfp:4 ingress egress
-

• 6，上传private binary

- 自己写的代码需要验证的时候需要用到private binary，而且是自己编译的
- 编译target下的image，并替换原先的LT上image
- NPOTB编译目录: /vobs/esam/build/npot-b/OS
- NPOTA编译目录: /vobs/esam/build/polt-b/OS
-
- Step1: 进入OS目录，launch cmake
- Step2: cmake ddl.XXX //为了区别出你的私有build和baseline不同，最好起一个特别的名字
- Step3: 修改NT板的/Sw的写权限: tapi enable write /Sw
- Step4: 备份一下NT上面LT原有的image
 - copy NDB4AA43.730 NDB4AA43.bak
- Step5: 替换image:
 - tftp 135.252.37.71(NT OAM地址_fox)
 - bin
 - put AAAA /Sw/AAAA
- Example:
 - tftp 135.252.37.71
 - tftp> bin
 - tftp> put NDB4AA43.sarah /Sw/NDB4AA43.863
 - Sent 3749475 bytes in 18.4 seconds
 - tftp> quit
- Step6: 重启LT板，重新登录到LT上，查看build name是否是自己编译的名称

• 7，Lab配置相关命令

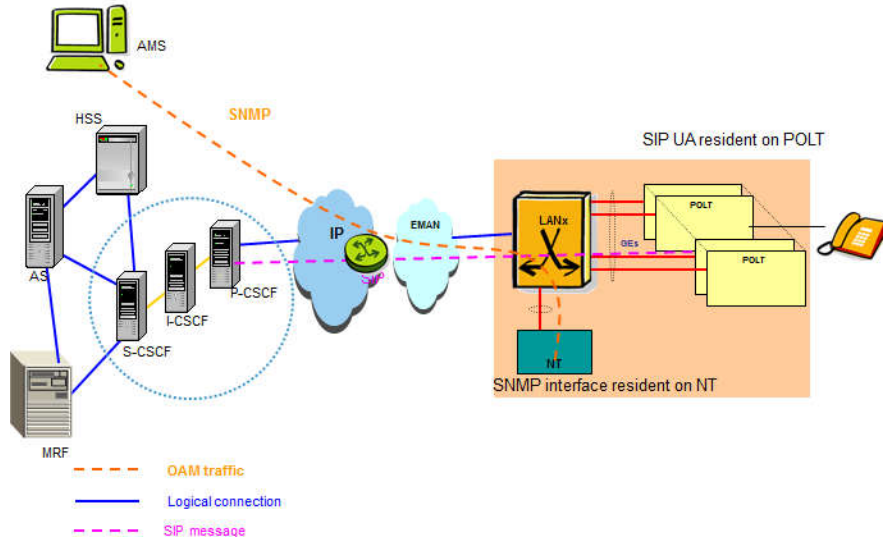
1，查看OAM IP地址和默认路由
info configure system mangement
2，查看tel状态
show voice sip termination
查看checkout的文件
ctl lsp -co
3，查看binary
show software-mngt oswp
4，查看NT和LT卡在什么槽位
show equipment slot
show interface shub port （更详细信息）
5，重启LT卡
admin equipment slot lt:1/1/16[CLI上显示的slot号] reboot with-selftest
重启NT卡
admin equipment reboot-isam without-self-test
6，octopus方式登录NT: 该方式需要在linux上进行，而不是直接通过连接软件
1. octopus STDIO xxx.xxx.xxx.xxx[OAM地址]:UDP:23
2. user:shell psw:nt
3. 登陆成功，正常操作
通过NT登陆LT板（抓取log时用到）
eqpt displayASAM
login board 11XX[eqpt displayasam出来的LT板编号]
有时候会出现ALL Task Busy的情况，这个时候输入
rcom exec -b 11XX -c login kill
使用octopus登陆带LT上之后主要是用dbg命令来抓取不同模块的log。抓取log的前提是mirror端口正确配置。
相关的octopus命令如下：
7，查看voice sip配置信息
info configure voice sip
8，删除某个sip termination
configure voice sip no termination 1/1/3/1

• 3，ISAMV系统架构

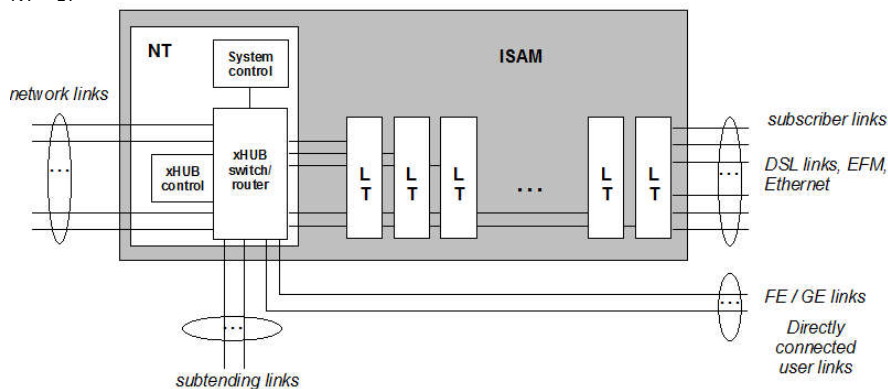
• 1, 硬件组成

- ISMAV在网络中的位置: 连接PSTN和IMS。

- POLT: 业务板卡, 直接连接用户电话机, 模拟信号和数字信号的编解码。相当于PC+DSP, 操作系统为Vxworks
- NT: ISAMV配置和管理



- NT和LT接口



NT SW

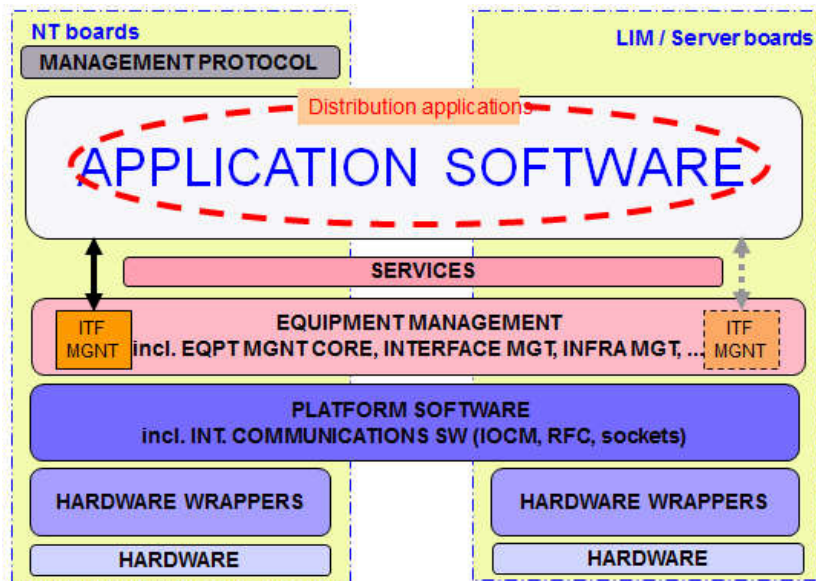
- **Persistent storage + all consistency checks (pre-provisioning!)** (IACM + xHUB)
- **OAM protocol stacks** (IACM + xHUB)
- **LANX control** (xHUB)
 - LANX configuration
 - LANX data plane control (L2, IP routing)
 - LANX control plane (IGMP, DHCP relay, ...)
- **Authentication (local or via Radius client)** (IACM, relay on SHUB)

LT SW (IACM)

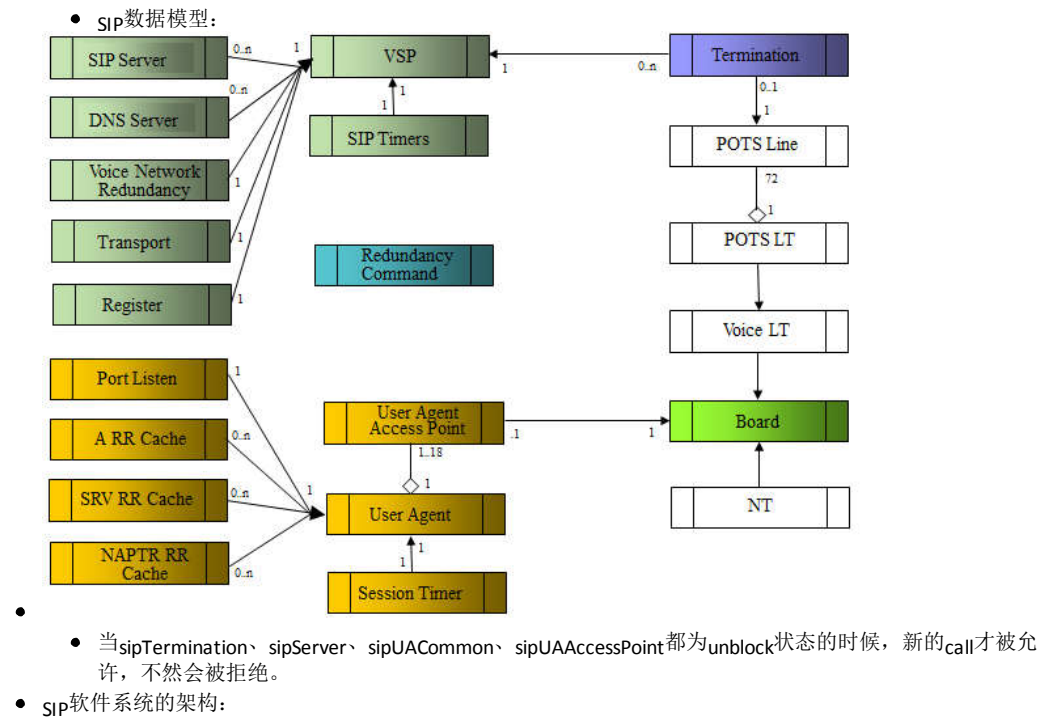
- **DSL / Ethernet chipset control** (DSL proxy / Ethernet HWW)
- **IWF control**
 - LT configuration
 - LT data plane control (L2) (e.g. assist MAC learning)
 - LT control plane (IGMP, 802.1x, DHCP/PPPoE relay, ARP proxy/relay, ...)
- **Statistics monitoring**

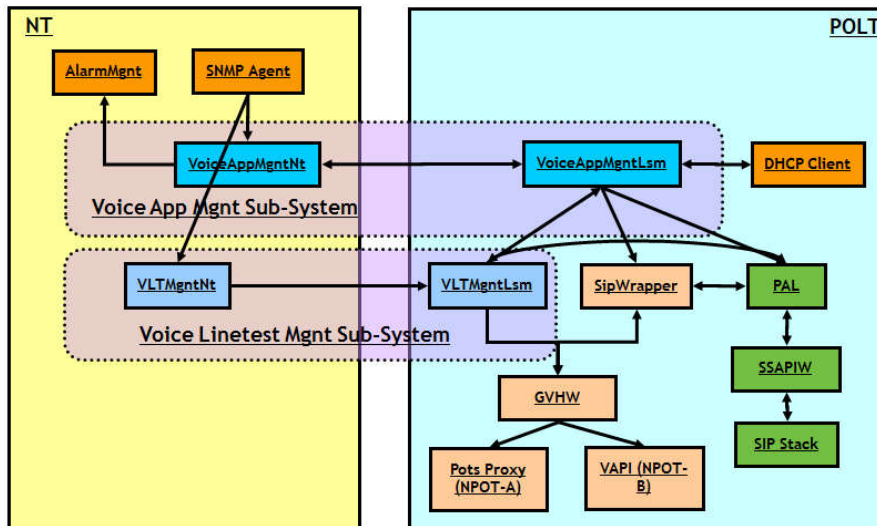
- NT卡的现有种类:

- EANT-A, ECNT-A, ECNT-C, NANT-A, AGNT-A, NANT-D



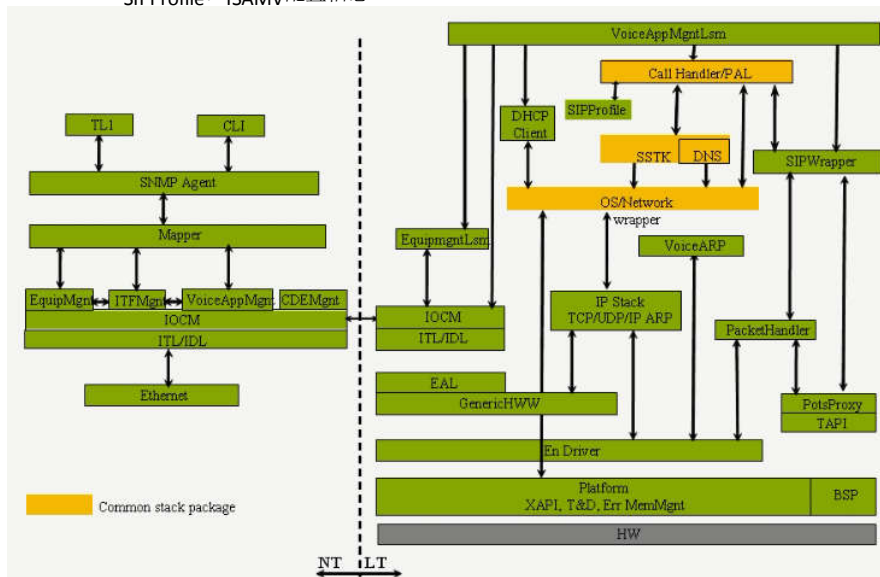
- 2, ISAM-V SIP系统架构

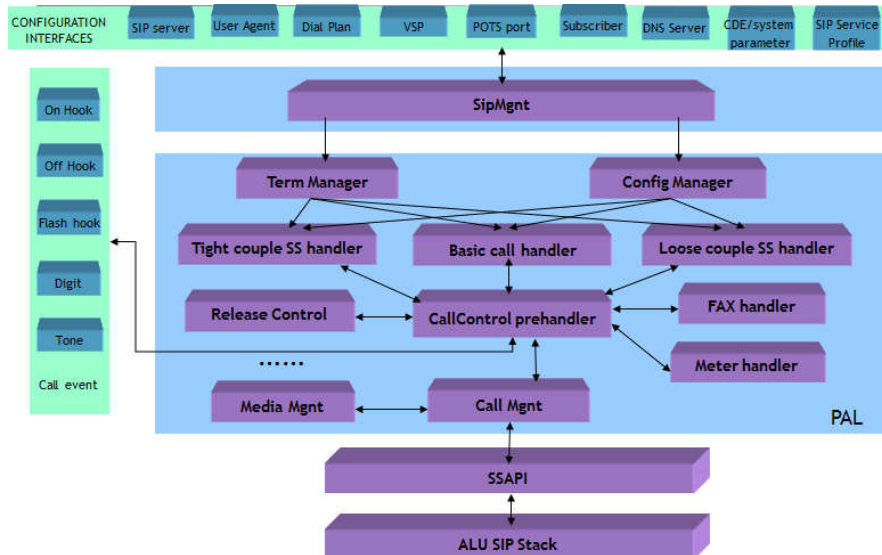




- **VoiceAppMgmtNt**: SIP管理的NT模块, 提供了SIP管理的SNMP接口。可以实现以下SIP管理: SIP server、SIP UA、Dial plan、Lineid syntax profile、VSP、SIP termination。可以加载和保存NT中上述管理的数据。处理SIP LT和AlarmMgmt之间的SIP警告。
- **VoiceAppMgmtLsm**: SIP管理的LSM模块, 从VoiceAppMgmtNt接收SIP的相关配置数据, 配置PAL、DHCPClient、IPStack、PacketHandler。启动DHCP Client; 从DHCPClient获取IP。将消息从PAL和SipWrapper传递到VoiceAppMgmtNt。设置L2和L3的vlan/ip参数。
- **VLTMgmtNt**: SIP线路测试的NT模块, 提供了SIP线路测试的SNMP接口。可以实现以下SIP线路测试的管理: session管理、line管理、parameters管理、report管理。处理VLTMgmtLsm的测试报告。
- **VLTMgmtLsm**: SIP线路测试的LSM模块, 从VLTMgmtNt接受测试命令并通知GVHW执行操作, 从VLTMgmtNt收集测试结果和报告。

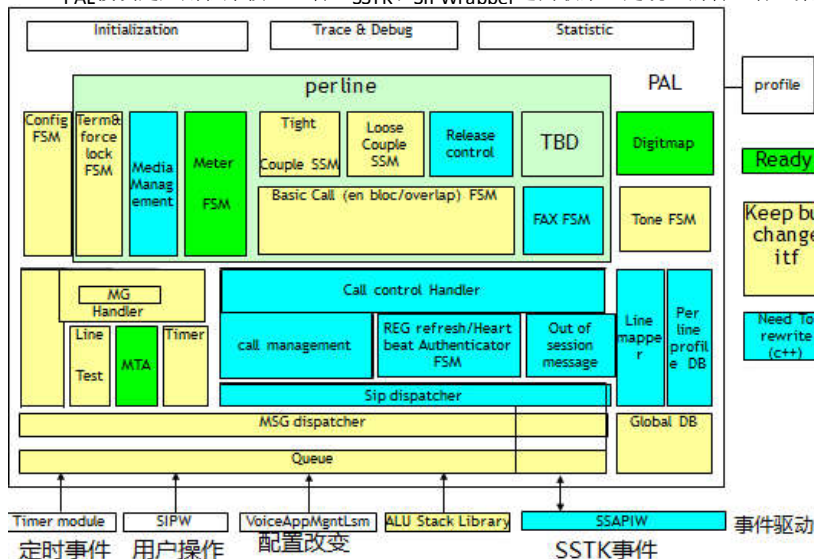
- NT、LT软件架构细节图, 其中黄色部分是最核心的部分, 也被成为CVP (comon SIP Voice package):
 - SIPWrapper: 管理具体的硬件, 包括DSP芯片, DSP资源管理(铃声), 电话机状态检测
 - SSTK: Sip Stack, 创建SIP消息并发送, 提供消息重传机制; 识别传入消息是否重传功能
 - PAL: 应用程序核心, 管理SSTK和SIPWrapper之间联系, 为用户提供接打电话功能。
 - SIPProfile: ISAMV配置信息





3, PAL

- PAL 模块是应用程序核心，管理SSTK和SIPWrapper之间联系，是统一的管理者。架构：



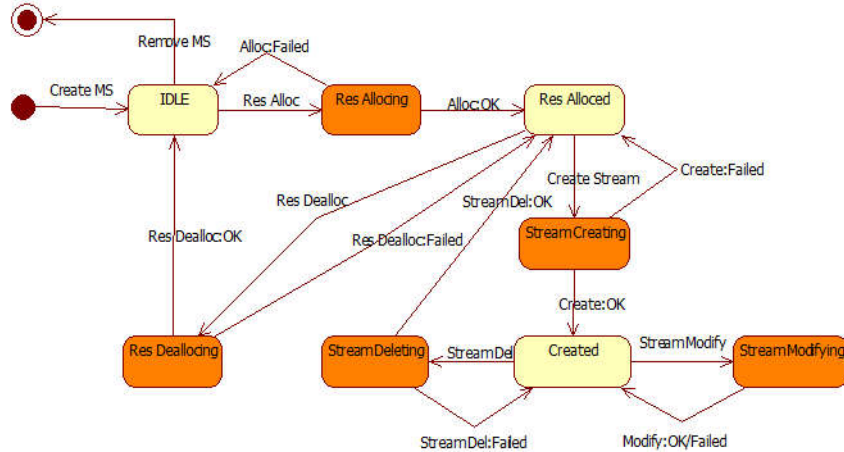
- PAL 中三个主要的内部模块：
 - CC: call control, 负责用户端的控制，与sipwrapper交互
 - CM: call management, 与Sip stack交互，负责网络侧信令的交互接发，管理资源和状态机状态
 - MM: media management, 负责具体资源的申请和释放；media stream的创建修改删除等；语音转换编解码等
 - Line mapper:
- 下面是一个拨打电话的流程，可以看到mm, cc, cm之间的合作关系：





- PAL是事件驱动，而且使用有限状态机，根据不同事件进行操作后进入某一个状态，不同的时间会导致不同的状态，比如注册和打电话的状态就不一样。下面列举的是MM资源分配的不同状态：

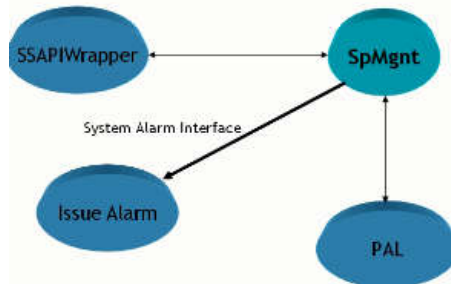
FSM State Chart



• 4, SipMgnt(VoiceAppMgntLsm)

• 1, SipMgnt成员

- SipMgnt属于CVP的一部分，只和下面的模块有接口交互：



The responsibilities of SipMgnt:

1) Data management:

Accept configuration requests from data provision layer.

Store all writeable objects and keep the logic relationships among them.

Enable those requests in SipMgnt/PAL/SSM modules.

2) Service management

PAL as the event-driven system working for call services.

SipMgnt as data-driven system working for management related services.

SipMgnt manages status of objects in ISAM-V (system status, sip server status, termination status, UA status ...)

RegisterFSM controlled by SipMgnt. (realize in SIP restoration)

SIP restoration in it.

Other management related services might be moved into it.

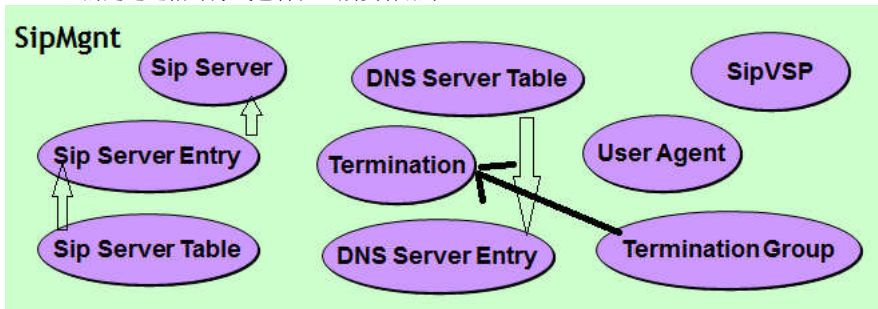
Interface between SSAPIWrapper

1. Send SIP Option Req/Resp
Move from PAL to SipMgnt
2. Send SIP Register Req/Resp
Move from PAL to SipMgnt
3. Configure ALU SIP stack Req/Resp (TcpConnection needed)
Move from PAL to SipMgnt (done)
4. Configure DNS server Req/Resp(TcpConnection needed)
New in SipMgnt
5. Perform DNS look-up Req/Resp
New in SipMgnt

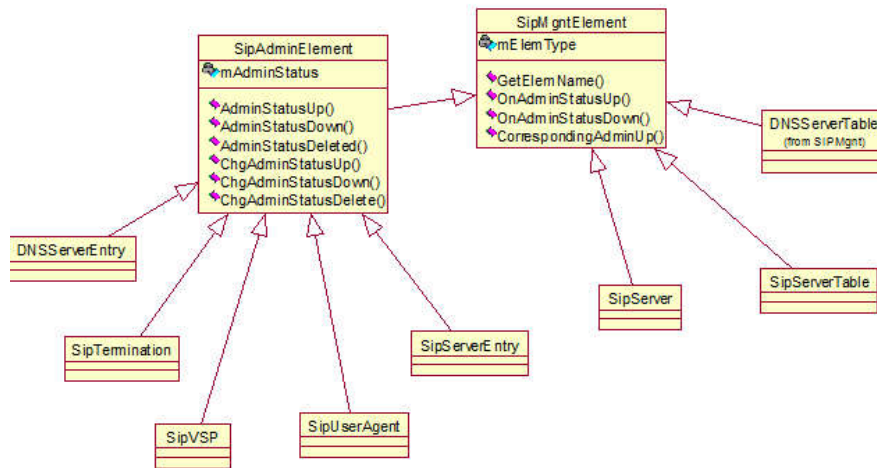
Interface between PAL

1. Enable/disable termination data in PAL
Need be enhanced
2. Query call status on termination, query pots line alarm status
Need be enhanced
3. PAL asks SipMgnt for some policies
Add by requirements
4. PAL reports service degradation(normal) events to SipMgnt
New in both SipMgnt and PAL
5. Handle Data provisioning common interface events
Done except Redundancy table

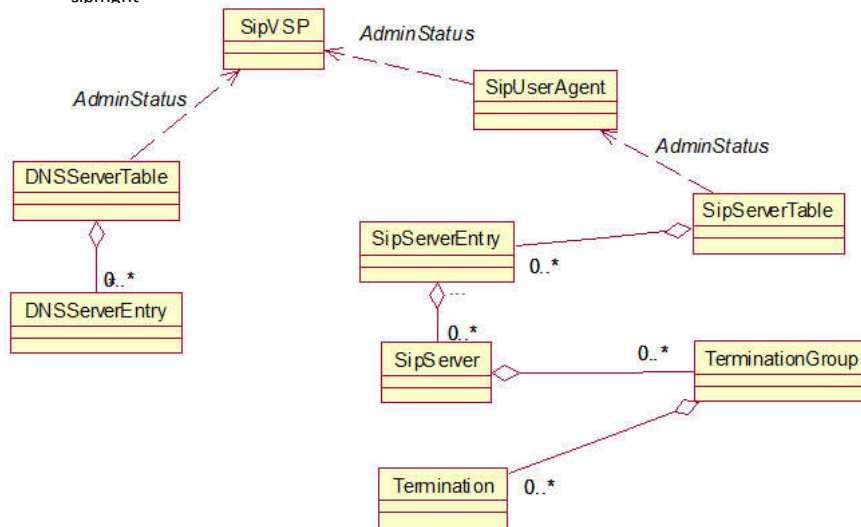
- SipMgnt主要包含处理和SIP相关的成员，这些成员在配置SIP lab的时候就会涉及到，各成员之间有箭头标记的是通过指针方式包含在上层类容器中：



- 上面SIP的成员都是从基类中继承来的，基类是SipAdminElement和SipMgntElement：



- 各sipmgt成员之间的关系如下:



List Operations Of Elements

SipUserAgent:

Configure ALU SIP stack

DNSServerTable:

Configure DNS servers to SIP stack

SIPServer:

Health monitoring to keep sip server status

Termination:

Do registration and de-registration

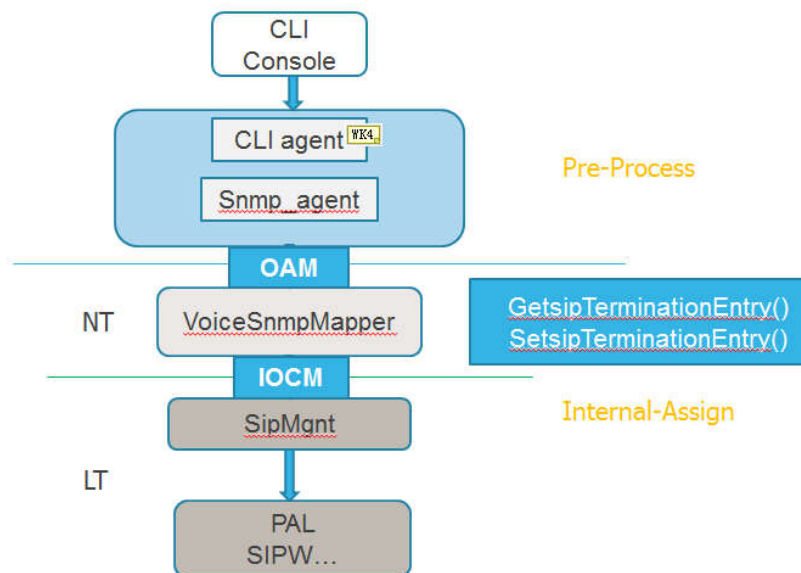
- 每个成员通过以下三种方式显示自身操作的结果: Ins、Oos、Unknown。成员通过event和FSM（状态机）进行操作的变化。

2, SipMgt Polling

- Polling是sip管理中timer的一种封装, 提供周期性的, 需要手动去停止。每一个termination中对应一个polling, 每一个polling只会专注一件事情。

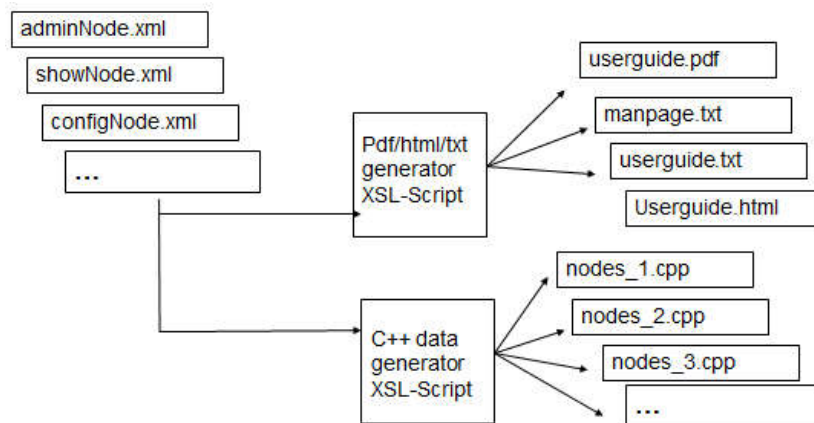
- 1) Polling provides a periodic check mechanism for elements in SipMgnt.
- 2) To derived from the base class, elements can define their own polling jobs.
- 3) In each polling job, you can:
 - 3.1) Adjust interval when you change the purpose of current polling
 - 3.2) Start/stop polling when needed
 - 3.3) Define the behaviors each time when polling timer expired
 - 3.4) Store context object in polling to perform the polling purpose
- Polling和timer的对比:
 - 1) Actually, polling is based on PAL' s timer system (genlib timer)
 - 2) In PAL, timer usually used only once at moment, and will be freed immediately after it expired
 - 3) In SipMgnt, we usually want a periodic check to wait some things happen, we won' t delete those polling when their owner exists
 - 4) Using polling, you don' t need define any timer event and structure, neither to find a message queue to dispatch timer to target objects.
- 5) Polling system has better OO manner
- 5, CLI&SNMP&VoiceAppMgntNt/Lsm

- ISAMV中CLI命令解析执行的过程:



1. CLI

- ISAMV可以使用独立的元素管理器（AMS或AWS）来管理，或者使用命令行选项直接管理节点，常用的命令行有：TL1、CLI。CLI语言是通过XML来进行结构的定义。CLI的常用命令分为以下几类：
 - New configure Command – 2PW/command
 - New show/admin command – 1 PW/command
 - Modify configure Command – 1PD/command
 - Modify show or admin command – 0.5 PD/command
- CLI最终是要实现对于MIB表里的表项的操作，一部分表项是read-write的，另一部分表项是read-only的。一般而言，read-write的表项应该写在configure命令里，read-only的表项应该写在show命令里。
- CLI的代码结构分为两部分：一是C++源代码，一是用户帮助文档，具体结构如下：



-
-
- 如何添加一个新的命令:
- CLI_Language/showNode.xml\ CLI_Language/adminNode.xml\ CLI_Language/configNode.xml
-
-



-
- 除了上面三种命令，还有两种不太常用的命令sample和poll
 - Example for "poll" command: source08/CLI_IP/ping.xml
 - Example for "sample" command: source08/CLI_IP/traceroute.xml
-
- SST: 对于新增的CLI命令的检查
 - For "show" commands – test the following
 - On-line help ("?" at the end of the command)
 - With res-id specified in all possible combinations (Eg., show equipment slot 1/1/4, show equipment slot nt-a, etc)
 - Without res-id specified (Eg., show equipment slot detail)
 - Tabular format – default (eg., show equipment slot)
 - Detailed format (eg. show equipment slot lt: 1/1/4 detail, show equipment slot detail etc)
 - XML format (eg. show equipment slot xml)
 - help <cli command> (eg. help show equipment slot xml)
 - tree <cli command> (eg. tree show equipment slot)
 - Any other command handler-types apart from typical "help" and "tree" that are supported by the show command
-
-

For "configure" command test the following

-
-
-
- On-line help ("?" at the end of the command)
- With res-id specified (all combinations)
- With parameter-types specified (all combinations)
- With all parameter-types that support "no" option
- info <cli command> in all 3 formats
- help <cli command>
- tree <cli command>
- Any other command handler-types apart from typical "help", "tree" and "info" that are supported by the configure command
-
-

For “admin” command, test the following:

- On-line help (“?” at the end of the command)
- With res-id specified (all combinations)
- With parameter-type specified (if applicable)
- help <cli command>
- tree <cli command>
- Any other command handler-types apart from typical “help” and “tree” are supported by the “admin” command

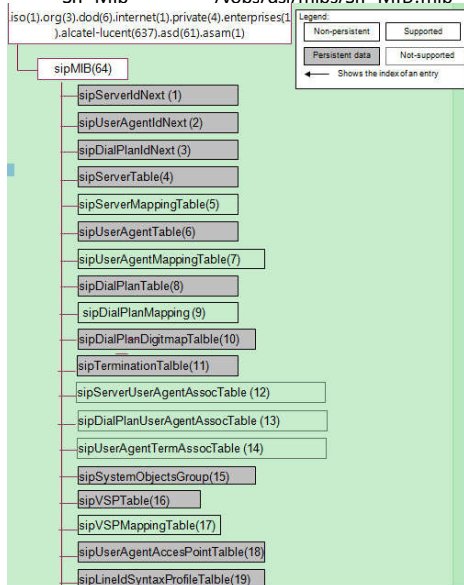
•
•
•
•
•
•

2. Mib

- Mib 全称 Management Information Base，意为管理信息数据库，定义了 ISAMV 上 SNMP 管理的接口。Mibs 一般被 SNMP 的管理器（AWS、AMS）使用，当然也被 TL1 和 CLI 的命令行使用。
- OID (对象标识符)，是 SNMP 代理提供的具有唯一标识的键值。MIB (管理信息基) 提供数字化 OID 到可读文本的映射。
- 管理信息库 (MIB, Management Information Base) 是 TCP/IP 网络管理协议标准框架的内容之一，MIB 定义了受管设备必须保存的数据项、允许对每个数据项进行的操作及其含义，即管理系统可访问的受管设备的控制和状态信息等数据变量都保存在 MIB 中

•

- SIP-Mib 的位置: /vobs/dsl/mibs/SIP-MIB.mib。Mib 的图形样式如下:



•

[BR29]

•

•

•

- Mib 是基于文本的，上面是其图形化的结构，下面讲解其真实的文本结构:
- RFC1213-MIB DEFINITIONS ::= BEGIN [BR30]
-
- IMPORTS [BR31]
- mgmt, NetworkAddress, IpAddress, Counter, Gauge, TimeTicks
- FROM RFC1155-SMI [BR32]
- OBJECT-TYPE
- FROM RFC 1212; [BR33]
-
- mib-2 OBJECT IDENTIFIER ::= { mgmt 1 } [BR34]
-
- -- groups in MIB-II

•

- system OBJECT IDENTIFIER ::= { mib-2 1 }
- interfaces OBJECT IDENTIFIER ::= { mib-2 2 }
- at OBJECT IDENTIFIER ::= { mib-2 3 }
- ip OBJECT IDENTIFIER ::= { mib-2 4 }
- icmp OBJECT IDENTIFIER ::= { mib-2 5 }
- tcp OBJECT IDENTIFIER ::= { mib-2 6 }
- udp OBJECT IDENTIFIER ::= { mib-2 7 }
- egp OBJECT IDENTIFIER ::= { mib-2 8 }
- transmission OBJECT IDENTIFIER ::= { mib-2 10 }
- snmp OBJECT IDENTIFIER ::= { mib-2 11 }[\[BR35\]](#)
-
- -- the Interfaces table
-
- -- The Interfaces table contains information on the entity's
- -- interfaces. Each interface is thought of as being
- -- attached to a 'subnetwork.' Note that this term should
- -- not be confused with 'subnet,' which refers to an
- -- addressing-partitioning scheme used in the Internet
- -- suite of protocols.
-
- ifTable OBJECT-TYPE[\[BR36\]](#)
- SYNTAX[\[BR37\]](#) SEQUENCE OF IfEntry
- ACCESS not-accessible [\[BR38\]](#)
- STATUS mandatory[\[BR39\]](#)
- DESCRIPTION
- "A list of interface entries. The number of entries is
- given by the value of ifNumber."
- [\[BR40\]](#) ::= { interfaces 2 } [\[BR41\]](#)
-
- ifEntry OBJECT-TYPE
- SYNTAX IfEntry
- ACCESS not-accessible
- STATUS mandatory
- DESCRIPTION
- "An interface entry containing objects at the subnetwork
- layer and below for a particular interface."
- INDEX { ifIndex }[\[BR42\]](#)
- ::= { ifTable 1 } *OID=1.3.6.1.2.1.2.2.1*
-
- IfEntry ::= *定义一张的表，注意首字母大写，区别于对象的定义*
- SEQUENCE {
- ifIndex
- INTEGER,
- ifDescr
- DisplayString,
- ifType
- INTEGER,
- ifMtu
- INTEGER,
- ifSpeed
- Gauge,
- ifPhysAddress
- PhysAddress,
- ifAdminStatus
- INTEGER,
- ifOperStatus
- INTEGER,
- ifLastChange
- TimeTicks,
- ifInOctets
- Counter,
- ifInUcastPkts
- Counter,
- ifInNUcastPkts
- Counter,
- ifInDiscards
- Counter,
- ifInErrors
- Counter,
- ifInUnknownProtos
- Counter,
- ifOutOctets
- Counter,

- ifOutUcastPkts
- Counter,
- ifOutNUcastPkts
- Counter,
- ifOutDiscards
- Counter,
- ifOutErrors
- Counter,
- ifOutQLen
- Gauge,
- ifSpecific
- OBJECT IDENTIFIER
- }
-
- ifIndex OBJECT-TYPE
- SYNTAX INTEGER[BR43]
- ACCESS read-only [BR44]
- STATUS mandatory
- DESCRIPTION
- "A unique value for each interface. Its value ranges
- between 1 and the value of ifNumber. The value for
- each interface must remain constant at least from one
- reinitialization of the entity's network management
- system to the next reinitialization."
- ::= { ifEntry 1 }
-
- ifDescr OBJECT-TYPE
- SYNTAX DisplayString (SIZE (0..255))
- ACCESS read-only
- STATUS mandatory
- DESCRIPTION
- "A textual string containing information about the
- interface. This string should include the name of
- the manufacturer, the product name, and the version
- of the hardware interface."
- ::= { ifEntry 2 }
-
- END[BR45]
-
-
-
- 如何修改Mib:
 1. 修改mib文件
 2. 修改相关联的CLI文件
 3. Build mib
 4. Build相关联的模块
 5. 测试代码
 6. 修改MIB和CLI的文档
 7. 检查代码
-

• 3, IOCM

- IOCM全称: 'inter-object communication manager', 是一个提供位置透明、平台独立、实时响应和控制的通信机制, 通常用于分布式、实时的嵌入式环境中。其优点有:
 1. high level communication across address space boundaries
 - task, processor, network address spaces
 2. minimise the dependency among software components
 - change management (versioning), reuse, maintenance
 3. support team-work during the whole development process
 - incremental, parallel, multi-site development, change management
 4. provide dynamic environment
 - reconfiguration, load balancing, redundancy management, notification
 5. support multiple platforms
 - different hardware architectures, operating systems, compilers
 6. support variety of communication facilities
 - standard and proprietary communication protocols and chips
 7. provide effective solutions
 - control on speed versus memory optimisation, generic services
 8. provide effective test environment
 - multi-card simulation, simulation while having connection to the target
-

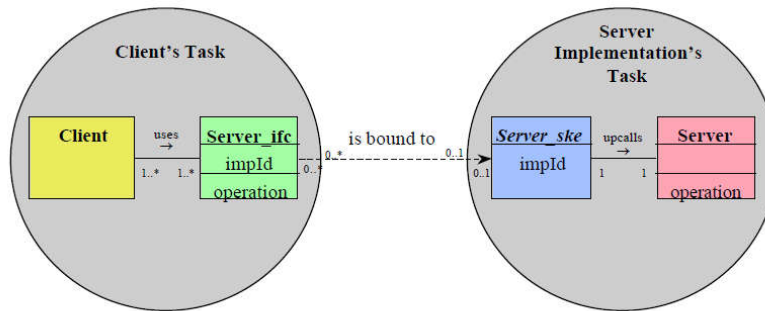


Figure 2: Location/platform transparent inter-object communication: the solution. Class diagram.

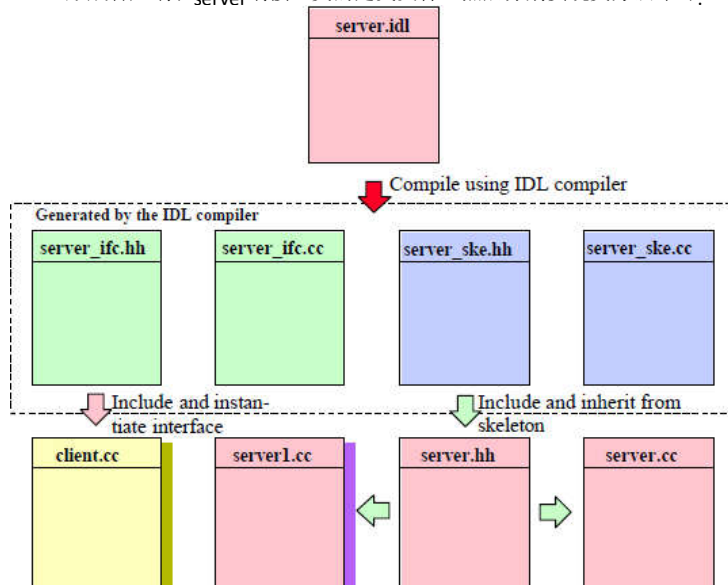
A client (e.g. an instance of `Client`) wants to use the services of a server (e.g. an instance of `Server`). To achieve this in a location transparent way, the client instantiates the interface class `Server_ifc`, that was provided by the server and binds the interface to the server with object id `impId`. From this point, the client just calls the server operations on that interface object as if the interface object was the server itself. The IOCM takes care of the rest.

Remark, at server side, the operation is not directly executed on the real server object but on a `Server_ske` instance (which keeps the object id) instead. The latter will finally upcall the server. This will be explained in detail in section 3.3. For the time being, consider the `Server_ske` and the `Server` instances as one and the same object (they have a 1-1 relationship anyway).

- IOCM接口的调用可以在同一模块内部的不同对象中，也可以在不同模块的对象之间。下面示例就是NT板内部和NT、LT之间IOCM调用的应用：



- IOCM使用IDL文件来定义接口，类似android中的AIDL。接口操作一般分为两类：request-only和request-response。对于request-only的类型需要使用oneway关键字区别。Idl文件经过IDL编译器编译之后就会分别形成客户端和server端接口类的定义文件，然后分别实现类方法即可。



- 下面介绍IDL文件的编写和编译：

1. IDL文件.idl

```

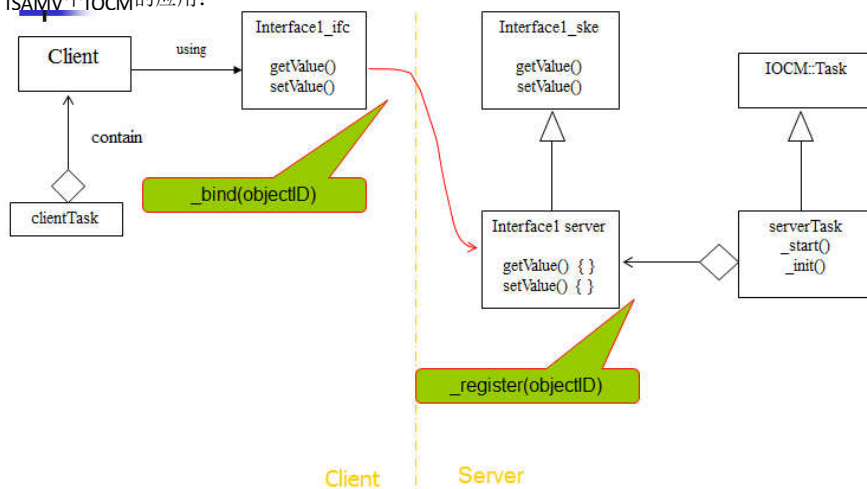
module module_name
{
    //data type definition[BR46]
    typedef unsigned long uint32;
    typedef unsigned short uint16;
    typedef char uint8;
    typedef sequence <char> T_string
}

//interface definition
interface interface_name

```

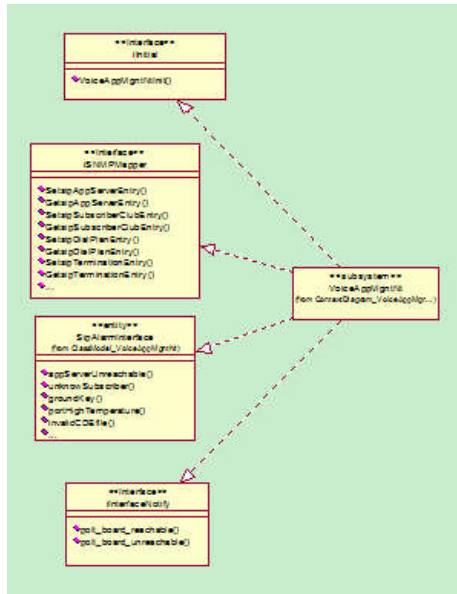
<pre> { void[BR47] func1(in para1, out para2, inout para3[BR48]); ... }; }; </pre>	
2. 编译idl文件 <ul style="list-style-type: none"> ◦ cmake M=LibIdl ◦ idlco /vobs/dsl/source08/IGMPMgntCommon/export/igmp.idl[BR49] ◦ 编译后自动生成如下文件，自动生成的不要进行修改： <ul style="list-style-type: none"> • Interface1_ifc.hh —— interface header file (for client inclusion) • interface1_ifc.cc —— interface implementation file (for client side) • interface1_ske.hh —— skeleton header file (for server inclusion) • interface1_ske.cc [BR50] —— skeleton implementation file (for server side) 	
3. client端不需要做什么操作，server需要新建一个类继承IDL中module下的接口方法： <ul style="list-style-type: none"> ◦ NewClass.hpp: ◦ class NewClass:public interface1_ske{[BR51] ◦ public: ◦ void func1(in para1, out para2, inout para3); ◦ } ◦ NewClass.cpp: ◦ Void NewClass::func1():[BR52] 	
4. client端调用接口中的方法之前需要保证server端对象已经注册到IOCM系统中，然后client端bind server端的objectId，然后通过接口调用方法。 <ul style="list-style-type: none"> ◦ <ul style="list-style-type: none"> ◦ // Create Object ID. ID of the object we want to communicate with ◦ ASAM_StructuredObjectId myObjId_l(AsamIds::domainIdItsc_c, Itsc::subsystemIdTestMgr_c, TestManager::classId, 0); ◦ ◦ mgrInterface._bind(myObjId);[BR53] ◦ Interface1_ifc interface(); ◦ Interface.func1(1,out1,2);[BR54] ◦ 上面的调用方式是同步方式，如果要异步调用，需要继承IOCM:InvocationAsync类代替默认的invocation（再研究）	

- ISAMV中IOCM的应用：

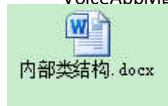


• 4，VoiceAppMgntNt

- SIP管理的NT模块，提供了SIP管理的SNMP接口。可以实现以下SIP管理：SIP server、SIP UA、Dial plan、Lineid syntax profile、VSP、SIP termination。可以加载和保存NT中上述管理的数据。处理SIP LT和AlarmMgnt之间的SIP警告。
- VoiceAppMgntNt的外部结构视图如下，包含初始化，SNMP命令操作，alarm和通知：

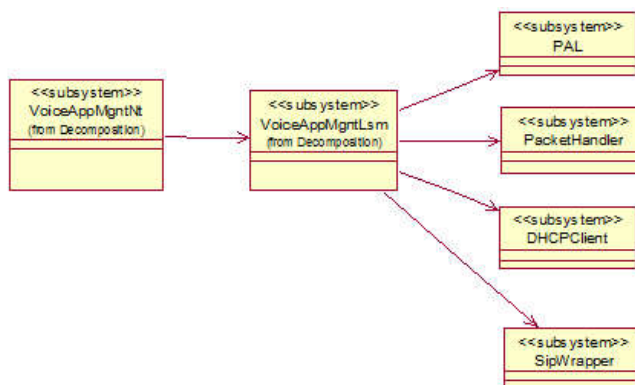


- VoiceAppMentNT内部类的结构视图:

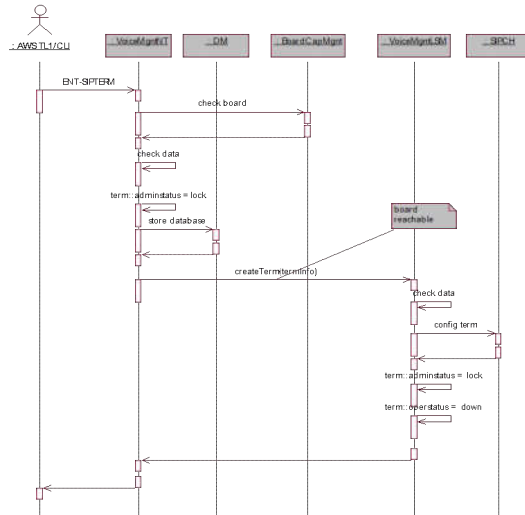


5, VoiceAppMgtLsm

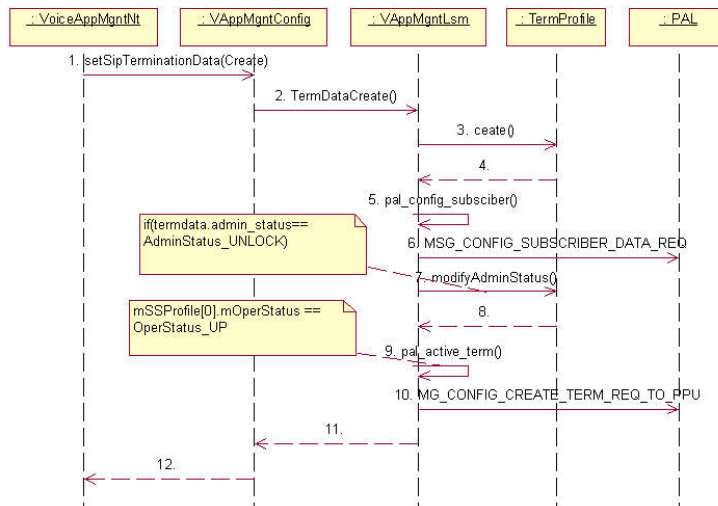
- 在LT板上的SipMgt, SIP管理的LSM模块, 从VoiceAppMgtNt接收SIP的相关配置数据, 配置sipServer、sipDialPlan、sipTermination、sipDialPlanDigitmp等。启动DHCP Client; 从DHCPClient获取IP。将消息从PAL和SipWrapper传递到VoiceAppMgtNt。设置L2和L3的vlan/IP参数。
- 下面列出的是和VoiceAppMgtLsm相关联的其他模块:



- CLI命令的执行流程, 涉及到NT和LT, 中间是使用IOCM机制进行通信:



-
-

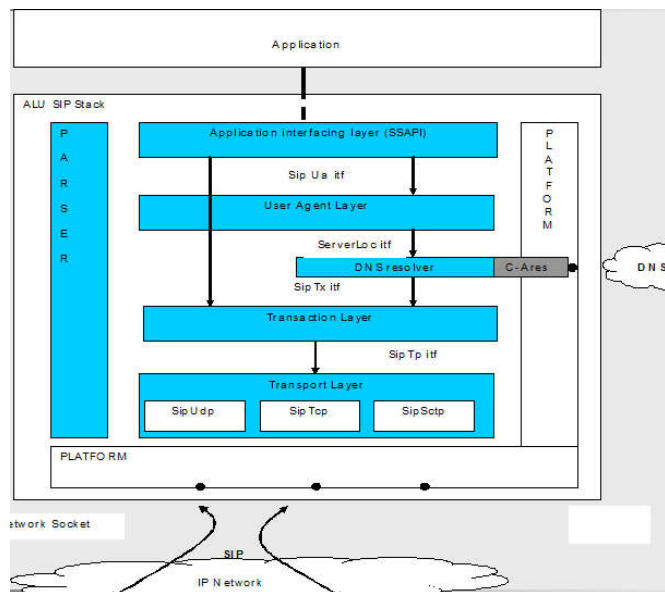


-
- 一个CLI命令 show voice sip termination 的执行流程:



6, SIP Stack

- SIP Stack结构如下, 基本和SIP协议结构一致。其中重要的几个部分的作用:
 - Parser: 对message进行解析
 - Platform: 从ip网络接收message
 - DNS resolver: DNS地址的解析
 - SSAPI: 与应用程序的接口, 用的最多的地方



- 在分析log的时候可以根据log的前缀来区分不同模块发出的log信息：比如SIPSSAPI就是SSAPI层的log：

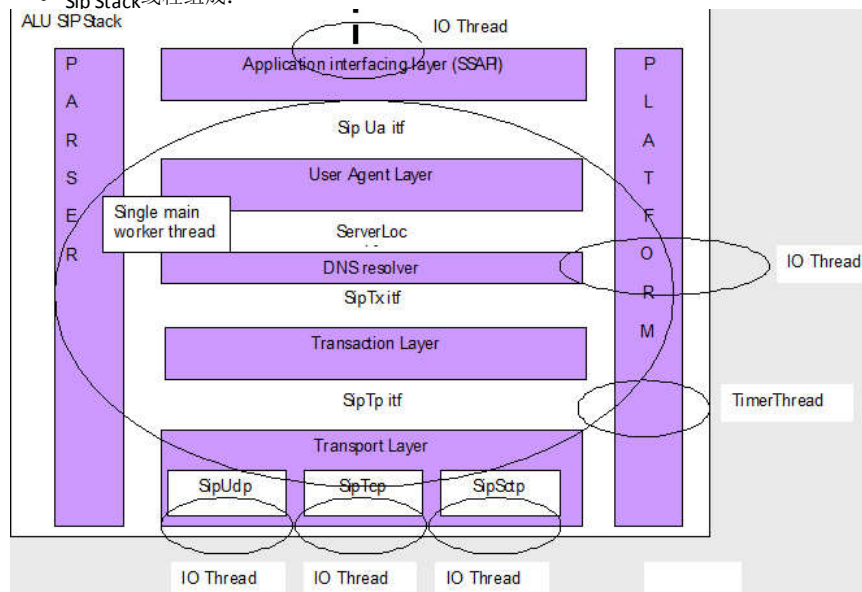
components

- sipssapi: module handling the TLV interface with the application.
- sipua : User Agent Layer module handling CallInstances (1 per call) and Dialogs (1 or more per CallInstance)
- siptx : Transaction Layer module
- sipudp : Transport layer module handling SIP over UDP
- ...

interfaces

- siptp : Interface definition of the Transport Layer (siptp_itf.h)
- siptx : Interface definition of the Transaction Layer (siptx_itf.h)
- sipua : Interface definition of the User Agent Layer (sipua_itf.h)
- siptypes : SIP Parser
- sdpcommon : SDP Parser

- Sip Stack线程组成:



- 1, SSAPI

- SSAPI是应用程序的接口，SSAPI可以接受应用（一般是PAL）的请求然后发送答复；同时应用要响应SSAPI的指令，两者之间的相互操作是异步进行的。SSAPI由message name（即消息ID）和message files的列表组成。一条消息的格式如下：

SyncPattern	Message Id	Message Length	Number of Attrs	Message Data (TLV like encoded)
<div></div>				
ID	L	VALUE	...	ID L VALUE

- SSAPI中的消息并不是规范的SIP消息，也不是最终发送和接收到的样子，而是通过SSAPI自身规定的属性（有必选和可选属性）和属性值组成的消息，经过解析或反解析生成最终的SIP标准消息。
- SSAPI根据层次不同可以划分为不同层次的消息，一般分为configuration、UA、transaction、DNS Resolver四个层次。每个层次都有不同类型的消息，不同的请求和响应也会通过不同层传入传出。

SSAPI消息：像SipMsg、Application Reference等都是必须有的属性
 : Invite_Request (103) ; len = 470,nbr of attrs = 3

```

Application_Reference: 30 31 32 63 2d 30 30 30 30 30 32 2d 30 65 32 66 6c 30
IpPortAddress: 31 37 32 2e 33 31 2e 32 30 31 2e 31 31 3a 38 30 36 30
SipMsg: INVITE sip:bert@172.31.128.127:5080 SIP/2.0
From: <sip:ua1@alcatel.be>
To: sip:bert@alcatel.be
Content-Type: application/sdp
Content-Length: 147

v=0
o=uac 1234567890 1234567890 IN IP4 172.31.225.44:5060
s=testcall
c=IN IP4 172.31.225.43
t=0 0
m=audio 10828 RTP/AVP 8
m=video 10830 RTP/AVP 31
  
```

SIP标准消息：是上面消息解析之后的样子

```

INVITE sip:bert@172.31.128.127:5080 SIP/2.0
Via: SIP/2.0/UDP 172.31.201.11:8060;branch=z9hG4bK*002e-00000001-0381
From: <sip:ua1@alcatel.be>;tag=0082-00000001-079f
To: sip:bert@alcatel.be
Call-ID: 0082-00000002-0b324b7e779c-50731e6bd-836fec8@sipua
CSeq: 1 INVITE
Max-Forwards: 30
Contact: sip:ua1@172.31.201.11:8060
Allow: INVITE, ACK, BYE, CANCEL, PRACK
Supported: 100rel
Content-Type: application/sdp
Content-Length: 147

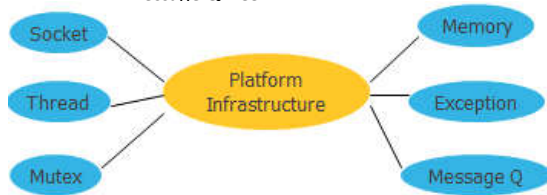
v=0
o=uac 1234567890 1234567890 IN IP4 172.31.225.44:5060
s=testcall
c=IN IP4 172.31.225.43
t=0 0
m=audio 10828 RTP/AVP 8
m=video 10830 RTP/AVP 31
  
```

- 在进行消息传递的时候，如果出现某些错误（消息无回应，返回消息错误等），SIPStack都会根据情况的不同生成消息通过SSAPI通知到应用，让应用知道当前的状态。这是SSTK的错误处理机制。

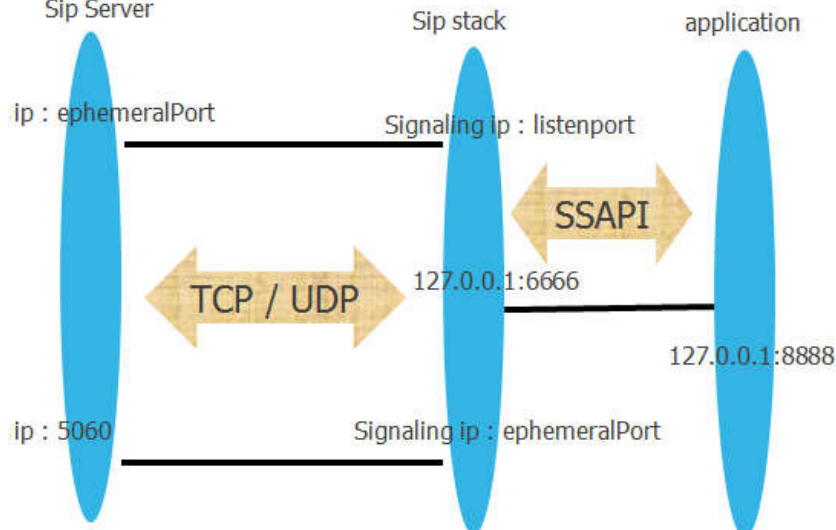
2、Platform

- Platform用来和SSTK外部通信，包括和IP网络通信传递SIP消息，和DNS通信进行域名转换等。Platform有下面几个基础组成：
 - Socket: 分不同平台提供统一的socket接口，用于和SIP server进行通信
 - Thread: 多线程异步处理

- **Mutex:** 互斥，提供了统一的调用接口，同时针对不同平台进行不同的实现
- **Memory:** 内存管理和内存池管理
- **Exception:** try-catch异常处理机制（可能缺少类库）。
- **Message Q:** SSAPI消息队列



- SSTK基本通信如下所示:
Sip Server



• 7, SIPWrapper

- SIPWrapper是介于PAL和GVHW之间的中间层，用于在两者之间传递消息。PAL通过Wrapper传递信令指令给硬件，GVHW通过wrapper传递事件给PAL。SIPWrapper的基本功能如下：

- Basic call & supplementary service
- Fax / Modem call (including VBD mode)
- SIP metering
- SIP line test
- SIP POTS statistics collection and report

- Interface to GVHW:
 - 事件event消息的格式如下:

Message ID (2 bytes)	Message Type (2 bytes) Not Used
Blank Area (4 bytes) Not Used	
Line ID (4 bytes)	
Event Type (4 bytes)	
Event (4 bytes)	
Extend Event(4 bytes)	

-
- Interface to PAL：
 - 消息类型为VBD
 ANS detect->switch to VBD->Finish,Resp to PAL

voipMsgHeader_t
mg_pkg_proc_evt_detected_ind_to_ppu_t
mg_oe_dtone_t

voipMsgHeader_t
mg_pkg_proc_change_to_vbd_req_from_ppu_t

voipMsgHeader_t
mg_pkg_proc_change_to_vbd_resp_to_ppu_t

-
- 另外SIPWrapper还与VappMgntLsm和CDE有接口进行通信。
-

附录:名词术语

POTS	老式普通电话服务
CLIP	Calling line Identification Presentation，主叫号码显示
CLIR	Calling Line Identification Restriction，主叫号码识别限制
MRF	Media Resource Function，媒体资源功能，用来提供媒体处理服务
SOC	系统级芯片
FLASH-Hook	话机上听筒放下去的位置上有一个凸出的方块按钮，称之为叉簧。一直按下去就是挂机，拿起听筒，叉簧凸出称为提机，用手拍一下称为“拍叉簧”。拍叉簧又叫flashhook，拍叉簧是一个将叉簧快速按下又放开的过程，本质上是切断直流通路大约80到200ms，拍一下叉簧，交换机不会认为是挂机，而是保持住这个呼叫，进行其他的一些操作；挂机是切断直流通路大约在200ms以上。拍"叉簧"典型的应用，就是进行电话转接(电话交换机普通分机转接，汇线通同组转接)。当需要把呼入的电话转到其他的分机时，本机拍一下插簧，然后进行转接，转接过程中电话交换机会保持住呼叫，直到转接成，重新进行连接。
CFU	Call Forwarding Unconditional，无条件呼叫转移
CFB	Call Forwarding Busy，遇忙呼叫转移
CFNR	Call Forwarding No Reply，无应答呼叫转移
NT/ LT	NT是控制板，控制多个LT版。用户先与NT板进行交互;LT是应用程序板子，有多个LT板子。LT版上部署了应用程序，比如CVP。
LANX	负责NT与LT之间的通信

OAM	Operation Administration and Maintenance, 操作管理维护
CLI	命令行界面, 路由器、终端复用器、调制解调器等界面
VxWorks	风河出品的实时操作系统
OSWP	应用程序的binary
CDE	配置文件, 里面会有一些参数的配置信息。系统软件会读取这个文件, 对系统中的参数进行配置, Customer Dependent Engineering
VRF	Virtual Routing Forwarding, VPN路由转发表
Voice vlan	Virtual Local Area Network, 虚拟局域网
FQDN	Fully Qualified Domain Name, 全称域名, 主机名+全路径
FS5K	SIP server
VSP	
Ins	
Oos	
Fallover	高优先级的sip server挂掉之后自动切换到低优先级的server
Fialback	高优先级的server恢复之后自动切换回高优先级server
ISDN	综合业务数字网, 在国内已经淘汰, 国外仍然在用
OID	对象标识符, object id
GHM	Health monitor, 分为foreground和background两种

-
-

已使用 Microsoft OneNote 2010 创建
一个用于存放所有笔记和信息的位置