//=============================================================

1,SM发送SM\_L2LINKEST\_IND消息给IWU，IWU接受到新消息之后发送MSG\_REPORT\_ISDN\_LINE\_INFO\_IND给MGMT

IWU：

iwuCommonEntryPoint(void)

| |🡪 isgProcessMsgStackMgr(msgBuf);

| | |🡪 retVal = isgCfgProcessMsg(rspMsgP, &error);

| | | |🡪 case ISG\_MODULE\_STKMNGR: retVal = isgCfgHandlePerIsdnLineReq(msgP, &buffered, ecodeP\_o);

| | | |🡪 retVal = isgCfgProcessMsgStackMgr(msgP, ecodeP\_o);

| | | | |🡪 retVal = getIptkLineidfromIsdnLineid(cfgGlobalData.isdnLineId, &cfgGlobalData.iptkLineId, ecodeP\_o);

| | | | |🡪 retVal = isgCfgFsmHandler(msgP, &intrnlHdr, ecodeP\_o);

| | | | | |🡪retVal = cfgCreateTcb(intrnlHdrP->transId, &(intrnlHdrP->tcbP), ecodeP\_o);

| | | | | | |🡪\*tcbP = (IsgCfgTcbSt \*) isgMemGet(sizeof(IsgCfgTcbSt));

| | | | | |🡪 retVal = isgCfgTcbEventHandler[intrnlHdrP->tcbP->cfgState][intrnlHdrP->event]( msgDataP, intrnlHdrP, ecodeP\_o);

isgCfgSmL2LinkEstInd(msgP\_i, intrnlHdrP\_io, ecodeP\_o)

|🡪 retVal = getIptkLineidfromIsdnLineid (intrnlHdrP\_io->tcbP->isdnLineId,&intrnlHdrP\_io->tcbP->iptkLineId, ecodeP\_o);

…..

|🡪 retVal=isgCfgReportIsdnLineInfoIndToMgmt(intrnlHdrP\_io->tcbP->isdnLineId, IFS\_UP, ecodeP\_o);

| |🡪 retVal = isgCfgSend(ISG\_MODULE\_MGMT, MSG\_REPORT\_ISDN\_LINE\_INFO\_IND, respMsgP, respMsgLen, ecodeP\_o);

| | |🡪 case ISG\_MODULE\_MGMT: retVal = isgSendToMgmt(apiId, msgP, msgLen);

|🡪 if (NOT\_REGISTERED == lineconfigdataP->registrationStatus)

|🡪 retVal = cfgDelTcb(intrnlHdrP\_io->transId, ecodeP\_o);

//======================================================================

2,VSIP接收到MSG\_REPORT\_ISDN\_LINE\_INFO\_IND消息

VSIP：

sipcch\_rcv\_pal\_tsk()

|🡪 case MSG\_REPORT\_ISDN\_LINE\_INFO\_IND: VAppMgntLsm::getinstance()->voip\_handle\_isdn\_line\_info\_ind((msg\_report\_isdn\_line\_info\_ind\_t \*)(pMsg+sizeof(voipMsgHeader\_t)));

| |🡪mLineProfile[pData->dLineId-PAL\_INC].modifyOperStatus(pData->isdnuStatus);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*通话流程开始\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//=======================================================================

3,Q931模块生成setup请求，发送给IWU模块内部的CC模块

Q931：

isdnl3\_process\_msg(buf\_t \* pbuffer,isdnl3\_error\_t \*perror)

|🡪 stat=(\*msg\_process\_routines[pbuffer[APID\_MSB]])(pbuffer,perror);

process\_l2\_msg (buf\_t \*pbuffer,isdnl3\_error\_t \*perror)

|🡪 stat=isdnl3\_handle\_msg\_l2\_l3 (pbuffer, p\_cstr\_buf, controller\_no,dlci\_val,perror) ;

| |🡪 if ((\*(\*p\_fsm\_table)[state][event\_id])(pbuffer,llci\_val, event\_id, p\_cstr\_buf, perror) == FAILURE)

isdnl3\_cmn00\_setup (pbuffer, llci\_val, p\_cstr\_buf, perror)

|🡪 if (send\_msg\_l3\_cc (pbuffer, p\_cstr\_buf, llci\_val, L3\_SETUP\_IND, perror) == FAILURE)

| |🡪 if (dispatch\_buffer (pbuffer) != SUCCESS)

| | |🡪 if(xq\_send(isdnCommonData\_g.IWUQid, msg\_buf)!= 0)

//=======================================================================

4,IWU CC模块收到setup请求，初始化call ，并向L3发送请求更多信息的请求，同时发送信令指示播放拨号音

IWU CC：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_Q931: if(ISG\_SUCCESS==isgProcessMsgQ931(msg\_i))

| | |🡪 if (ISG\_SUCCESS == isgCcProcessMsg(msg\_i, &error))

| | | |🡪 if(ISG\_FAILURE== (retVal=isgCcL3BufferToL3CS((IsgU8bit\*)ccMsgP\_i->isgData,l3MsgP,errorCode\_o)))

| | | | |🡪 retVal=isdnl3\_primitive\_convert\_bstocs(l3Msgp\_o, ccMsgP\_i,ETSI,NETWORK,L3\_BRI,errorCode\_o);

| | | |🡪 retVal = isgCcGetCallContextFromLlci(l3MsgP->llci,&(isgCcGlobalData.currentCallContext));

……

| | | |🡪 retVal=isgCcAllocateCallContextFromLlci(isdnLlci,&ccCallContext)

| | | | |🡪 if((ISG\_SUCCESS==isgCcContextCreateAndInit(ccCallContextPP\_o))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

Call context内容为：

[[========CC CallContext=========

Call ID:0 [IWU ]---

Llci ID:99 [IWU ]---

ISDN LineID: 0 [IWU ]---

IPTK LineID: 0[IWU ]---

CC Call State: CcCallState\_IDLE[IWU ]---

Sub State Flag: 0x0[IWU ]---

Media State: IwuMediaState\_MEDIA\_UNINITIATED[IWU ]---

Bearer Channel Id: 0[IWU ]---

Digits Received: (none)[IWU ]---

toneAplied: 0[IWU ]---

============================

| | | |🡪retVal=isgCcFsm [ccCallContext->currentCallState][ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪 isgCcReleaseL3CS(l3MsgP);

isgCcProcessL3SetupIndication\_CcState\_IDLE(IsgMsgSt \*ccMsgP\_i, IsgErrorEt \*errorCode\_o)

|🡪 retVal=isgCcAllocateBch(ccCallContext,errorCode\_o);

| |🡪 if(ISG\_SUCCESS==(retVal= isgIrmAllocResReq(bearerChannelId,&ccCallContext\_i->bearerChannelId, altAccept, ISG\_IRM\_CONNECTED,errorCode\_o)))

|🡪 retVal=isgCcAnalyseDigitsFromL3(ccCallContext,isgIwuLineDataP,errorCode\_o);

|🡪retVal=isgCcInitiateCallSetUp(ccCallContext,errorCode\_o);

| |🡪 if(subsData->registrationStatus==NOT\_REGISTERED)

| |🡪 ccCallContext\_i->internalCallId=isgCcGenerateCallId();

| |🡪 retVal=isgCcConvertApiQ931ToQ931 (L3\_SETUP\_IND\_TO\_CC\_MORE\_INFO\_REQ,ccCallContext\_i, &l3RespMsgP,errorCode\_o);

| |🡪 retVal=isgCcSendL3Msg((l3\_msg\_st\*)l3RespMsgP,ccCallContext\_i,errorCode\_o);

| | |🡪 retVal=isdnl3\_send\_unpack\_to\_l3 (CC\_ID,l3MsgP\_i,1024,pbuffer,ETSI,USER,L3\_BRI,(isdnl3\_error\_t\*)errorCode\_o);

| | | |🡪 returnvalue = dispatch\_buffer(pbuffer);

| |🡪 retVal = isgCcSaveL3Msg(ccCallContext\_i);

| |🡪

|🡪 mra\_fn\_sendStartSignalBasic(ccCallContext->isdnLlci, ccCallContext->bearerChannelId,ccCallContext->connectedIptkLineId,MG\_S\_DIAL\_TONE,IsgT302AtimerDuration\_g, IsgCcToneDuration,ccCallContext->internalCallId,errorCode\_o);

| |🡪 if (mra\_fn\_GetCallContextUsingCallId(&mraLineProfile[dLineId],

(IsgS32bit)dCallId, &(mraLineProfile[dLineId].pCurrentCallContext), &dError) == ISG\_FAILURE)

| |🡪 if (mra\_create\_context(dLineId, dCallId, scn\_term\_id) == ISG\_FAILURE)

| | |🡪 mra\_fn\_sendResourceAllocRequest(dTransactionId, scn\_term\_id, &dError);

| | | |🡪 isgSendToSipWrap(dVoipMsg.msg\_id, pMsgReq\_voip, (IsgU32bit)dMsgLen);

| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id,pMsgReq\_voip,dMsgLen);

|🡪 retVal=isgCcStartTimer(ISG\_CC\_TIMER\_T\_302\_A, ccCallContext->internalCallId, &ccCallContext->ccTimerData.timerId, errorCode\_o);

| |🡪 isgStartTimer(timerInfo, duration,timedIdP\_o);

//===================================================================

5,SIPW接收到资源分配的请求，开始处理,返回消息给IWU

SIPW：

handle\_res\_alloc\_req\_from\_ppu(unsigned long msg\_buf\_id,voipMsgHeader\_t \*p\_msg)

|🡪 send\_msg\_to\_ppu((unsigned long)p\_resp, p\_resp->length);

handle\_start\_signal\_req\_from\_ppu(unsigned long msg\_buf\_id,voipMsgHeader\_t \*p\_msg)

|🡪 send\_msg\_to\_ppu((unsigned long)p\_resp, p\_resp->length);

//========================================================================

6，IWU接收到SIPW发回资源分配和播放开始音的回复消息，进行处理

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_RES\_ALLOC\_RESP\_TO\_PPU: (\*mra\_glb\_mmHandlerArray [(mraProfile->pCurrentCallContext->dMediaState).dCurrentState]) (mraProfile, pCallData,Isg\_en\_resAllocRespRecv, &dErr);

mra\_fn\_mm\_waitForPlatRespons

|🡪case Isg\_en\_resAllocRespRecv:

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_PKG\_PROC\_START\_SIGNAL\_RESP\_TO\_PPU:

//====================================================================

7,Q931模块根据平台上报事件生成information请求，发送给IWU模块内部的CC模块

Q931：

isdnl3\_process\_msg(buf\_t \* pbuffer,isdnl3\_error\_t \*perror)

|🡪 stat=(\*msg\_process\_routines[pbuffer[APID\_MSB]])(pbuffer,perror);

process\_l2\_msg (buf\_t \*pbuffer,isdnl3\_error\_t \*perror)

|🡪 stat=isdnl3\_handle\_msg\_l2\_l3 (pbuffer, p\_cstr\_buf, controller\_no,dlci\_val,perror) ;

| |🡪 if ((\*(\*p\_fsm\_table)[state][event\_id])(pbuffer,llci\_val, event\_id, p\_cstr\_buf, perror) == FAILURE)

isdnl3\_cmn00\_setup (pbuffer, llci\_val, p\_cstr\_buf, perror)

|🡪 if (send\_msg\_l3\_cc (pbuffer, p\_cstr\_buf, llci\_val, L3\_SETUP\_IND, perror) == FAILURE)

| |🡪 if (dispatch\_buffer (pbuffer) != SUCCESS)

| | |🡪 if(xq\_send(isdnCommonData\_g.IWUQid, msg\_buf)!= 0)

//========================================================================

8,IWU CC收到Q931发送的消息，开始分析号码是否有效，并请求停止通话开始音

IWU CC：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_Q931: if(ISG\_SUCCESS==isgProcessMsgQ931(msg\_i))

| | |🡪 if (ISG\_SUCCESS == isgCcProcessMsg(msg\_i, &error))

| | | |🡪 if(ISG\_FAILURE== (retVal=isgCcL3BufferToL3CS((IsgU8bit\*)ccMsgP\_i->isgData,l3MsgP,errorCode\_o)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm [ccCallContext->currentCallState][ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪 isgCcReleaseL3CS(l3MsgP);

isgCcProcessL3InfoIndication\_CcState\_OVERLAP\_SENDING(IsgMsgSt \*ccMsgP\_i, IsgErrorEt \*errorCode\_o)

|🡪 retVal=isgCcAnalyseDigitsFromL3(ccCallContext,isgIwuLineDataP,errorCode\_o);

|🡪 retVal=isgDaStartMonitorDigitMap(dGlbTaskId,dGlbDigitmapId ,

isgIwuLineDataP->enterKey, l3MsgP->controller\_no, (IsgU16bit\*)&ccCallContext->dmlTransId, (dl\_error\_t \*)errorCode\_o);

| |🡪 dRetVal = dl\_start\_monitor\_digitmap(task\_id, digitmap\_id, pGlbParsedDigitMap[term\_id].pDigitmapValue, term\_id, p\_transc\_id,\ (dl\_error\_t \*)p\_ecode);

|🡪 retVal=isgDaTraverseDigitMap(ccCallContext->dmlTransId, ccCallContext-> digitsReceived,(dl\_error\_t\*)errorCode\_o);

| |🡪 retVal = dl\_traverse\_digit\_map( transc\_id, digit, (dl\_error\_t \*)p\_ecode);

| | |🡪 num\_dm\_ptrs = dl\_check\_digit\_in\_dm(transc\_id, &temp\_pat\_complete,&temp\_complete,&temp\_timer\_type,&success,p\_temp\_dm\_stk,digit,&p\_digitmap\_stk,num\_dm\_ptrs,counter,&modifier\_flag,&immediate\_mod\_flag,p\_ecode);

| | |🡪 dl\_report\_digit\_comp\_event(transc\_id,DL\_PARTIAL\_TIMEOUT);

| | | |🡪 (\*g\_dl\_comp\_table[g\_dl\_trans\_list[transc\_id].task\_id])(transc\_id, g\_dl\_trans\_list[transc\_id].task\_id, g\_dl\_trans\_list[transc\_id].term\_id, p\_timestamp,&digit\_data);

|🡪 isgDaStopMonitorDigitMap( ccCallContext->dmlTransId,(dl\_error\_t\*) errorCode\_o);

isgCcProcessDmlNormalCall( IsgU16bit txnid\_i, IsgU32bit service\_id\_i, IsgU32bit index\_i, gl\_time\_stamp\_ut \*time\_stamp\_i, dl\_digit\_data\_t\* digitData\_i)

|🡪 mra\_fn\_sendStopSignalRequest(ccCallContext->isdnLlci,ccCallContext->bearerChannelId, ccCallContext->connectedIptkLineId,MG\_S\_DIAL\_TONE,ccCallContext->internalCallId,&errorCode);

| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id,pMsgReq\_voip,dMsgLen);

| |🡪 mra\_fn\_sendResourceDeallocRequest (mra\_getMmPlatformTxnId(mraProfile), scn\_term\_id,pErr);

| | |🡪 isgSendToSipWrap(dVoipMsg.msg\_id, pMsgReq\_voip, (IsgU32bit)dMsgLen);

|🡪 case DL\_PARTIAL\_MATCH :

//========================================================================

9，IWU接收到SIPW发回停止开始音和资源释放的回复消息，进行处理

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_PKG\_PROC\_STOP\_SIGNAL\_RESP\_TO\_PPU:

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_RES\_DEALLOC\_RESP\_TO\_PPU: (\*mra\_glb\_mmHandlerArray [(mraProfile->pCurrentCallContext->dMediaState).dCurrentState]) (mraProfile, pCallData, Isg\_en\_resDeallocRespRecv, &dErr);

mra\_fn\_mm\_waitForPlatRespons

|🡪case Isg\_en\_resDeallocRespRecv:

//==========================================================================

10, Q931模块根据平台上报事件继续生成information请求，发送给IWU模块内部的CC模块，直到所有号码拨号完毕

IWU CC:

|……

| | |🡪 dl\_report\_digit\_comp\_event(transc\_id,DL\_PARTIAL\_TIMEOUT);

| | | |🡪 (\*g\_dl\_comp\_table[g\_dl\_trans\_list[transc\_id].task\_id])(transc\_id, g\_dl\_trans\_list[transc\_id].task\_id, g\_dl\_trans\_list[transc\_id].term\_id, p\_timestamp,&digit\_data);

|🡪 isgDaStopMonitorDigitMap( ccCallContext->dmlTransId,(dl\_error\_t\*) errorCode\_o);

isgCcProcessDmlNormalCall( IsgU16bit txnid\_i, IsgU32bit service\_id\_i, IsgU32bit index\_i, gl\_time\_stamp\_ut \*time\_stamp\_i, dl\_digit\_data\_t\* digitData\_i)

|🡪 case DL\_UNAMBIGOUS\_MATCH: isgCcEncodeXml (ccCallContext,&ccCallContext->isgCcXmlBufP,&errorCode)

|🡪 retVal=pclConvertApiQ931ToIptk (L3\_SETUP\_IND\_TO\_ICF\_CREATE\_CALL\_REQ , isgCcPclConvpar, l3MsgP,(icf\_msg\_st\*\*)&iptkMsgP,&errorCode);

| |🡪 case L3\_SETUP\_IND\_TO\_ICF\_CREATE\_CALL\_REQ: retVal = pclConvISDNSetupToCreateCallReq(convParam\_i, q931MsgStP\_i, icfCreateCallMsgP, errorP\_o);

| | |🡪 retVal = getIsgIwuSystemProfile(&sysProfile, errorP\_o);

| | |🡪 retVal = getIsgIwuLineConfigData(&subsProfileP, q931MsgP\_i->controller\_no,ISDN\_LINE\_ID, errorP\_o);

| | |🡪 retVal = pclConvertBearerCapIE(bCapP, hlcP, & icfCreateCallMsgP\_io->p\_pref\_media\_profi e, errorP\_o);

|🡪 if(ISG\_SUCCESS==(retVal=isgCcSendIptkMsg((icf\_msg\_st\*)iptkMsgP, ccCallContext,&errorCode)))

| |🡪 retVal=isgSendToIPTKFromIcfMsg(iptkMsgP\_i->hdr.api\_id, iptkMsgP\_i, iptkMsgP\_i->hdr.api\_length);

|🡪 retVal=isgCcStartTimer(ISG\_CC\_TIMER\_T\_INFO, ccCallContext->internalCallId, &ccCallContext->ccTimerData.timerId,&errorCode);

| |🡪 isgStartTimer(timerInfo, duration,timedIdP\_o);

//======================================================================

11,IPTK接收到创建会话的请求，开始处理。IPTK与SIP server进行信令的交互和处理，发送ICF\_GET\_CODEC\_LIST\_REQ给IWU

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessIcfGetCodecListReq\_CcState\_MediaSetup\_in\_progress

|🡪 retVal=mra\_process\_cc\_msg(pRecvdIptkMsg,( IsgVoidP \*) &icfMsgGetCodecRespP, ISG\_NULL);

| |🡪 case ICF\_GET\_CODEC\_LIST\_REQ:if (ISG\_FAILURE == (\*mra\_glb\_mmHandlerArray [dCurrentMMState]) (&mraLineProfile[dLineId], pCallData, Isg\_en\_getCodecList, &dError))

|🡪 retVal=isgCcEncodeXmlFromCodecListResp(icfMsgGetCodecRespP,ccCallContext, ( IsgU8bit\*\*)&ccCallContext->isgCcXmlBufP,errorCode\_o);

|🡪 if(retVal) retVal=isgCcSendIptkMsg(icfMsgGetCodecRespP,ccCallContext,errorCode\_o);

mra\_fn\_mm\_idle(mraProfile, pCallData, dInputEvent, pErr)

|🡪 case Isg\_en\_getCodecList: retVal = mra\_fn\_sendCodecListResp(mraProfile->dLineId,

(IsgU32bit)pCallContext->dCallId, ISG\_SUCCESS, &pAcceptedStream, &rejected\_stream, pErr);

//======================================================================

11,IPTK接收到ICF\_GET\_CODEC\_LIST\_REQ的回复，开始处理。之后发送ICF\_OPEN\_MEDIA\_CHANNEL给IWU

IWU:

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessIcfOpenMediaChannelReq\_CcState\_MediaSetup\_in\_progress

|🡪 retVal=mra\_process\_cc\_msg(pRecvdIptkMsg, (IsgVoidP \*)&icfOpenMediaChRespP, ccCallContext->bearerChannelId);

| |🡪 case ICF\_OPEN\_MEDIA\_CHANNEL\_REQ: (\*mra\_glb\_mmHandlerArray[dCurrentMMState])

(&mraLineProfile[dLineId], pCallData, Isg\_en\_openMediaChannelReq, &dError);

mra\_fn\_mm\_idle(mraProfile, pCallData, dInputEvent, pErr)

|🡪 case Isg\_en\_openMediaChannelReq: retVal = mra\_fn\_sendResourceAllocRequest (mra\_getMmPlatformTxnId(mraProfile), mraProfile->pCurrentCallContext->dChannel, pErr);

| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id, pMsgReq\_voip, (IsgU32bit)dMsgLen);

//======================================================================

12,SIPW接收到分配资源的请求之后进行相应的处理，然后发送回复消息给IWU MM：

MG\_RES\_ALLOC\_RESP\_TO\_PPU

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_RES\_ALLOC\_RESP\_TO\_PPU: (\*mra\_glb\_mmHandlerArray [(mraProfile->pCurrentCallContext->dMediaState).dCurrentState]) (mraProfile, pCallData,Isg\_en\_resAllocRespRecv, &dErr);

mra\_fn\_mm\_waitForPlatRespons

|🡪case Isg\_en\_resAllocRespRecv: mra\_fn\_openMediaChannelResp(mraProfile->dLineId,

(IsgU32bit)pCallContext->dCallId, ISG\_SUCCESS, pErr);

| |🡪 isgCcProcessMraMsg(pMsg,dMsgLen,pErr);

| | |🡪 case ICF\_OPEN\_MEDIA\_CHANNEL\_RESP：retVal=isgCcSendIptkMsg(pRecvdIptkMsg, ccCallContext,errorCodeP\_o);

//======================================================================

13,IPTK接收到媒体信道打开的回复之后，进行处理。之后发送创建媒体会话的请求ICF\_CREATE\_MEDIA\_SESSION\_REQ

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessIcfCreateMediaSessionReq\_CcState\_MediaSetup\_in\_progress

|🡪 retVal=mra\_process\_cc\_msg(pRecvdIptkMsg, ISG\_NULL, ccCallContext->bearerChannelId);

| |🡪 case ICF\_CREATE\_MEDIA\_SESSION\_REQ: (\*mra\_glb\_mmHandlerArray[dCurrentMMState]) (&mraLineProfile[dLineId], pCallData, Isg\_en\_openMediaChannelReq, &dError);

mra\_fn\_mm\_idle(mraProfile, pCallData, dInputEvent, pErr)

|🡪case Isg\_en\_createMediaSessionReq: mra\_fn\_sendStreamCreateOrModifyRequest (mra\_getMmPlatformTxnId(mraProfile), mraProfile->pCurrentCallContext->dChannel, MG\_STREAM\_CREATE\_REQ\_FROM\_PPU, pFirstLocalStream->mode, pFirstLocalStream->stream\_type, &(pFirstLocalStream->local\_sdp), &(pFirstLocalStream->remote\_sdp),pErr);

| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id, pMsgReq\_voip, dMsgLen);

//=======================================================================

14,SIPW接收到MM发送的创建媒体流请求，进行处理。之后发送创建成功的回复：MG\_STREAM\_CREATE\_RESP\_TO\_PPU

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_STREAM\_CREATE\_RESP\_TO\_PPU: (\*mra\_glb\_mmHandlerArray [(mraProfile->pCurrentCallContext->dMediaState).dCurrentState]) (mraProfile, pCallData,Isg\_en\_resAllocRespRecv, &dErr);

mra\_fn\_mm\_waitForPlatRespons

|🡪 case Isg\_en\_streamCreateRespRecv: mra\_fn\_sendResourceTopoRequest (mra\_getMmPlatformTxnId(mraProfile), (U32)mraProfile->pCurrentCallContext->dChannel, (U32)mraProfile->pCurrentCallContext->dChannel, MG\_TPLGY\_BIDIRECT, pErr);

| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id, pMsgReq\_voip, dMsgLen);

//=======================================================================

14,SIPW接收到MM发送的Topo请求，进行处理。之后发送创建成功的回复：MG\_RES\_TOPO\_RESP\_TO\_PPU

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_RES\_TOPO\_RESP\_TO\_PPU: (\*mra\_glb\_mmHandlerArray [(mraProfile->pCurrentCallContext->dMediaState).dCurrentState]) (mraProfile, pCallData,Isg\_en\_resAllocRespRecv, &dErr)

mra\_fn\_mm\_waitForPlatResponse

|🡪case Isg\_en\_resTopoRespRecv: if(mra\_fn\_createMediaSessionResp(mraProfile->dLineId, (IsgU32bit)pCallContext->dCallId, ISG\_SUCCESS,pErr)==ISG\_FAILURE)

| |🡪isgCcProcessMraMsg(pMsg, dMsgLen, pErr);

| | |🡪 case ICF\_CREATE\_MEDIA\_SESSION\_RESP: retVal=isgCcSendIptkMsg(pRecvdIptkMsg, ccCallContext, errorCodeP\_o);

//=========================================================================

15,IPTK接收到媒体会话创建成功的回复之后，开始向sip server发送invite请求创建SIP会话，invite是通过SIPU模块创建，log中以PRINTING SIP PDU OF LENGTH: 1847----开头。

之后SIPU收到100 trying 临时回复和180 ringing回复，与iptk进行交互处理。Iptk将180 振铃的回复发送给IWU CC进行处理：ICF\_REMOTE\_USER\_ALERTED。

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessIcfRemoteuserAlertedInd\_CcState\_AnyState

|🡪 mra\_fn\_sendStartSignalBasic(ccCallContext->isdnLlci, ccCallContext->bearerChannelId, ccCallContext->connectedIptkLineId,MG\_S\_RING\_BACK\_TONE, MG\_TIMEOUT, IsgCcToneDuration,ccCallContext->internalCallId,errorCode\_o);

| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id,pMsgReq\_voip,dMsgLen);

|🡪 retVal=pclConvertApiIptkToQ931(ICF\_REMOTE\_USER\_ALERTED\_TO\_CC\_ALERTING\_REQ, convParam,pRecvdIptkMsg,&isdnMsgList ,errorCode\_o);

|🡪 isgCcSendL3Msg((l3\_msg\_st\*)isdnMsgList->q931BufP, ccCallContext,errorCode\_o);

//==========================================================================

16, SIPW处理播放呼叫音的请求,并发送MG\_PKG\_PROC\_START\_SIGNAL\_RESP\_TO\_PPU回复给IWU。同时Q931处理CC发送的消息，并将消息发送给L2层。

IWU处理SIPW发送的消息。

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_PKG\_PROC\_START\_SIGNAL\_RESP\_TO\_PPU:

//===========================================================================

17,SIPU收到200 ok消息，交给IPTK处理，IPTK决定连接通话，于是SIPW发送ack消息给Sip Server。Iptk发送ICF\_CODEC\_COMMIT\_REQ给IWU处理。

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessIcfCodecCommitReq\_CcState\_MediaSetup\_in\_progress

|🡪 retVal=mra\_process\_cc\_msg(pRecvdIptkMsg,( IsgVoidP \*) &icfCodecCommitRespP, ISG\_NULL);

| |🡪 case ICF\_CODEC\_COMMIT\_REQ: if (ISG\_FAILURE == (\*mra\_glb\_mmHandlerArray[dCurrentMMState]) (&mraLineProfile[dLineId], pCallData, Isg\_en\_commitReq, &dError))

|🡪 retVal=isgCcSendIptkMsg(icfCodecCommitRespP,ccCallContext,errorCode\_o);

mra\_fn\_mm\_idle

|🡪 case Isg\_en\_commitReq: mra\_fn\_sendCodecCommitResp(mraProfile->dLineId, (IsgU32bit)pCallContext->dCallId, pCodecCommitReq->p\_stream\_commit\_list, ISG\_SUCCESS,pErr))

//=====================================================================

18, IPTK收到codec commit回复之后，进行处理，之后发送修改媒体流请求给IWU：ICF\_MODIFY\_MEDIA\_SESSION\_REQ

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessModifyMediaSessionReq\_CcState\_Anystate

|🡪 retVal=mra\_process\_cc\_msg(pRecvdIptkMsg, ISG\_NULL, ccCallContext->bearerChannelId);

| |🡪 case ICF\_MODIFY\_MEDIA\_SESSION\_REQ:(\*mra\_glb\_mmHandlerArray [dCurrentMMState] )\ (&mraLineProfile[dLineId], pCallData, Isg\_en\_modifyMediaSessionReq, &dError);

mra\_fn\_mm\_idle

|🡪 case Isg\_en\_modifyMediaSessionReq: mra\_fn\_deleteSdpStreamClonedList (&(pLocalStream->remote\_sdp));

|🡪 mra\_fn\_cloneSdpStreamList(&pReceivedStream->sdp\_stream, &pLocalStream ->remote\_sdp);

|🡪 mra\_fn\_sendStreamCreateOrModifyRequest (mra\_getMmPlatformTxnId(mraProfile), mraProfile->pCurrentCallContext->dChannel, MG\_STREAM\_CREATE\_REQ\_FROM\_PPU, pFirstLocalStream->mode, pFirstLocalStream->stream\_type, &(pFirstLocalStream->local\_sdp), &(pFirstLocalStream->remote\_sdp),pErr);

| |🡪| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id, pMsgReq\_voip, dMsgLen);

//==========================================================================

19,SIPW收到修改媒体会话的请求，进行处理，之后发送回复给IWU：MG\_STREAM\_MODIFY\_RESP\_TO\_PPU

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_STREAM\_MODIFY\_RESP\_TO\_PPU: (\*mra\_glb\_mmHandlerArray [(mraProfile->pCurrentCallContext->dMediaState).dCurrentState]) (mraProfile, pCallData, dMmFsmEvent, &dErr);

mra\_fn\_mm\_waitForPlatResponse

|🡪 case Isg\_en\_streamModifyRespRecv: if(mra\_fn\_modifyMediaSessionResp (mraProfile->dLineId, (IsgU32bit)pCallContext->dCallId, ISG\_SUCCESS,pErr) == ISG\_FAILURE)

|……

| |🡪 isgCcProcessMraMsg(pMsg,dMsgLen,pErr);

| | |🡪 isgCcPrintCallContext(ccCallContext);

| | |🡪 case ICF\_MODIFY\_MEDIA\_SESSION\_RESP: isgCcSendIptkMsg(pRecvdIptkMsg, ccCallContext,errorCodeP\_o);

| | |🡪 isgCcPrintCallContext(ccCallContext);

//=====================================================================

20,IPTK收到修改媒体会话的回复，进行处理并通知SIPU正式接通通话。同时发送接通通话的消息给IWU处理：ICF\_CONNECT\_IND。

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessIcfConnectInd\_CcState\_AnyState

|🡪 mra\_fn\_sendStopSignalRequest(ccCallContext->isdnLlci,ccCallContext->bearerChannelId, ccCallContext->connectedIptkLineId,MG\_S\_RING\_BACK\_TONE,ccCallContext->internalCallId,errorCode\_o);

| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id,pMsgReq\_voip,dMsgLen);

|🡪 retVal=pclConvertApiIptkToQ931 (ICF\_CONNECT\_IND\_TO\_CC\_SETUP\_RESP, convParam, (icf\_msg\_st \*)pRecvdIptkMsg,&isdnMsgList ,errorCode\_o);

|🡪 isgCcSendL3Msg((l3\_msg\_st\*)isdnMsgList->q931BufP, ccCallContext,errorCode\_o);

|🡪 isgIrmSetResStatus(ccCallContext->bearerChannelId, ISG\_IRM\_ACTIVE,errorCode\_o);

//=======================================================================

21,SIPW收到停止呼叫音的请求，停止呼叫音，发送回复。Q931收到connect消息，通知L2层进行处理，并发送回复给IWU。

IWU：（SIPW）

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_PKG\_PROC\_STOP\_SIGNAL\_RESP\_TO\_PPU:

IWU：（Q931）

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_Q931: if(ISG\_SUCCESS==isgProcessMsgQ931(msg\_i))

| | |🡪 case REDUNDANCY:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/ /

/ 通话进行中 /

/ /

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//===========================================================================

22,主叫端用户挂断电话，Q931发送disconnect消息给IWU。IWU处理后向IPTK发送消息

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_Q931: if(ISG\_SUCCESS==isgProcessMsgQ931(msg\_i))

| | |🡪 default：if (ISG\_SUCCESS == isgCcProcessMsg(msg\_i, &error))

| | | |🡪 if(ISG\_FAILURE== (retVal=isgCcL3BufferToL3CS((IsgU8bit\*) ccMsgP\_i->isgData,l3MsgP,errorCode\_o)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm [ccCallContext-> currentCallState] [ccCallEvent] (ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪 isgCcReleaseL3CS(l3MsgP);

isgCcProcessL3GenericCallRelease\_CcState\_Anystate

|🡪 case L3\_DISC\_IND: retVal=pclConvertApiQ931ToIptk ( L3\_DISCONNECTION\_IND\_TO\_ICF\_TERMINATE\_CALL\_REQ, convParam,l3MsgP, &icfMsgP, errorCode\_o);

| |🡪 case L3\_DISCONNECTION\_IND\_TO\_ICF\_TERMINATE\_CALL\_REQ: retVal = pclISDNDisconnectToTerminateReq(convParam\_i, q931MsgStP\_i, icfTerminateCallMsgP, errorP\_o);

|🡪 retVal=isgCcSendIptkMsg(icfMsgP,ccCallContext,errorCode\_o);

//===================================================================

23,IPTK通知SIPU挂断电话，SIPU发送Bye SIP消息给SIPServer，然后通知IPTK消息已经发送。IPTK发送删除媒体会话的消息给IWU：ICF\_DELETE\_MEDIA\_SESSION\_REQ

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessDeleteMediaSessionReq\_CcState\_AnyState

|🡪 retVal=mra\_process\_cc\_msg(pRecvdIptkMsg, ISG\_NULL, ccCallContext->bearerChannelId);

| |🡪 case ICF\_DELETE\_MEDIA\_SESSION\_REQ: (\*mra\_glb\_mmHandlerArray[dCurrentMMState]) (&mraLineProfile[dLineId], pCallData, Isg\_en\_deleteMediaSessionReq, &dError);

mra\_fn\_mm\_idle

|🡪 case Isg\_en\_deleteMediaSessionReq: retVal = mra\_fn\_sendStreamDeleteRequest (dTransactionId,(U32)mraProfile->pCurrentCallContext->dChannel, dTransactionId, pErr);

| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id, pMsgReq\_voip, dMsgLen);

//===========================================================================

24,SIPW删除媒体会话，发送回复给IWU：MG\_STREAM\_DELETE\_RESP\_TO\_PPU

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_STREAM\_DELETE\_RESP\_TO\_PPU: (\*mra\_glb\_mmHandlerArray [(mraProfile->pCurrentCallContext->dMediaState).dCurrentState]) (mraProfile, pCallData, Isg\_en\_streamDeleteRespRecv, &dErr);

mra\_fn\_mm\_waitForPlatResponse

|🡪 case Isg\_en\_streamDeleteRespRecv: retVal = mra\_fn\_sendResourceDeallocRequest (mra\_getMmPlatformTxnId(mraProfile), pCallContext->dChannel, pErr);

| |🡪 isgSendToSipWrap(dVoipMsg.msg\_id, pMsgReq\_voip, (IsgU32bit)dMsgLen);

//===========================================================================

25,SIPW释放资源，回复IWU: MG\_RES\_DEALLOC\_RESP\_TO\_PPU

IWU:

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_MGMT: if(ISG\_SUCCESS==isgProcessMsgMgmt(msg\_i))

| | |🡪 if(ISG\_SUCCESS == isgProcessMsgSipWrap(msg\_i))

| | | |🡪 if (ISG\_SUCCESS == mra\_fn\_voipManagerMsgHandler(msg\_i))

| | | | |🡪 case MG\_RES\_DEALLOC\_RESP\_TO\_PPU: (\*mra\_glb\_mmHandlerArray [(mraProfile->pCurrentCallContext->dMediaState).dCurrentState]) (mraProfile, pCallData, Isg\_en\_streamDeleteRespRecv, &dErr);

mra\_fn\_mm\_waitForPlatResponse

|🡪 case Isg\_en\_resDeallocRespRecv: mra\_fn\_deleteMediaSessionResp((IsgU32bit) pCallContext->dCallId, ISG\_TRUE, pErr);

| |🡪 retVal = isgCcProcessMraMsg(pMsg, dMsgLen, pErr);

| | |🡪 retVal = isgCcProcessMraMsg(pMsg, dMsgLen, pErr);

//===========================================================================

25,SIPU收到Bye的回复200 ok，通知IPTK。IPTK收到IWU发送的删除媒体会话的回复，处理后发送ICF\_REL\_RES\_REQ给IWU。

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessIcfRelResourceReq\_CcState\_AnyState

|🡪 retVal=mra\_process\_cc\_msg(pRecvdIptkMsg,( IsgVoidP \*)&relResRespP, ISG\_NULL);

| |🡪 case ICF\_REL\_RES\_REQ: (\*mra\_glb\_mmHandlerArray[dCurrentMMState]) (&mraLineProfile[dLineId], pCallData, Isg\_en\_releaseResourceReq, &dError);

mra\_fn\_mm\_idle

|🡪 case Isg\_en\_releaseResourceReq: retVal = mra\_fn\_removeCallContextFromTermProfile (mraProfile, pCallContext->dCallId, pErr);

//========================================================================

26,IPTK发送ICF\_TERMINATE\_CALL\_RESP消息给IWU，终结通话

IWU：

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_IPTK: if(ISG\_SUCCESS==isgProcessMsgIPTK(msg\_i))

| | |🡪 if (ISG\_FAILURE == isgCommonDecodeIPTKmsg(msg\_i, &decodedMsgP))

| | |🡪if (ISG\_SUCCESS == isgCcProcessMsg(internalMsg, &error))

| | | |🡪 if(ISG\_FAILURE==(retVal=isgCcConvertApiToEvent

(ccMsgP\_i->isgHeader.orig\_mod\_id,api\_id,&ccCallEvent,eventStr)))

| | | |🡪 isgCcPrintCallContext(ccCallContext);

| | | |🡪retVal=isgCcFsm[ccCallContext->currentCallState] [ccCallEvent](ccMsgP\_i,errorCode\_o);

| | | |🡪 isgCcPrintCallContext(ccCallContext);

isgCcProcessIcfTerminateCallIResp\_CcState\_REMOTE\_TERMINATING

|🡪 isgCcSendL3Msg((l3\_msg\_st\*)isdnMsgList, ccCallContext,errorCode\_o);

|🡪 isgCcRemoveContextWithMaps(&ccCallContext);

//======================================================================

27,Q931处理之后发送回复给IWU：L3\_RELEASE\_CONFIRMATION

IWU:

iwuCommonEntryPoint(void)

|🡪 if (ISG\_FAILURE == isgProcessMsg(dataBufP))

| |🡪 case ISG\_MODULE\_Q931: if(ISG\_SUCCESS==isgProcessMsgQ931(msg\_i))

| | |🡪 if (ISG\_SUCCESS == isgCcProcessMsg(msg\_i, &error))

| | | |🡪 if(ISG\_FAILURE== (retVal=isgCcL3BufferToL3CS((IsgU8bit\*)ccMsgP\_i->isgData,l3MsgP,errorCode\_o)))

| | | |🡪 isgCcReleaseL3CS(l3MsgP);