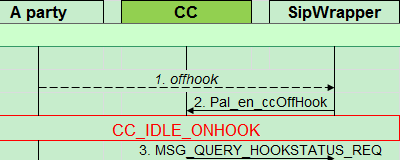
## 1，主叫端流程

### 1，1~3



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SIP Wrapper\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Root()

|--🡪application\_init()

| |-🡪init\_application()

| | |-🡪AccessNeZIPS\* pZips = new AccessNeZIPS (queueId);

|-🡪init\_SipWrapper

|-🡪xt\_start(task\_id, T\_PREEMPT|T\_TSLICE, SipWrapper\_main, NULL);

|-🡪SipWrapper\_main

handle\_HptMsg();

|-🡪Case POTS\_EVENT\_MSG:

|----🡪 Case GEN\_HOOK\_STATE\_CHANGED

|---🡪 handle\_term\_ev\_offhook()

| ---🡪 sipw\_led\_on() //Turn led on

|----🡪 sipw\_confDCSlope/// assign dsp resource

|---🡪 sipw\_assignDsp() //

|---🡪 sipw\_allocDspRes() // Get a free DSP resource

|----🡪 gen\_voice\_hww::getgen\_voice\_hww()->initiateCall(…..) // Allocates the resources related with basic call

|--🡪 gen\_voice\_hww::getgen\_voice\_hww()->configureDataChannel() //

|--🡪 send\_event\_to\_ppu(MG\_E\_OFF\_HOOK,termId);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter PAL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Function pal\_fn\_msghandler() is the key function for PAL. PAL module is event driven.

\* While(1) {} in pal\_fn\_msghandler to handle event/message(internal and external msg).

\* 1. Enter pal\_fn\_voipManagerMsgHandler to deal with term event and change term state

\* machine if there are term state change(e.g. off hook, on hook).

\* 2 then, pass msg to CC or MM by pal\_fn\_internalDerivedCcEvent and change CC or CM state

\* machine.

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

pal\_main

|

|-🡪pal\_fn\_msghandler

|-🡪 pal\_fn\_voipManagerMsgHandler

|

|--🡪 case MSG\_EVT\_DETECTED\_IND:

| |

| |-> case case MG\_E\_OFF\_HOOK:

| | |🡪pal\_fn\_sendLineFeedingRequest

| | | |->pal\_fn\_voipSendRequestToPlatform

| | | | |🡪 pal\_fn\_print\_msg\_to\_platform

| |

| | |🡪 dCurrentState =(pTermProfile->dTerminalState).dCurrentState; (\*pal\_glb\_stateHandlerArray[dCurrentState])(pTermProfile, g\_pCallData, Pal\_en\_offHook, &dErr);

pal\_fn\_idleOnHook() // This function handles the events received in IDLE State

|

|

|🡪 case Pal\_en\_offHook:

| |-> AllocLmId(LMID\_BSM\_ALLOC, (Pal\_ty\_u32bit \*)&pCallContextData->dCallId, pTermProfile->dLineId);

| |-🡪 pal\_fn\_printCallStatistics(pTermProfile, pCallContextData, pErr);

| |----🡪 pal\_internal\_msg\_mgr::getInstance()->post\_plat\_message\_on\_term

| | |-🡪 pal\_internal\_msg\_mgr::getInstance()->post\_plat\_message

| | | |-🡪 post\_message(orgFsm, dstFsm, event, lineId, ctxId, pmsg, isUrgent);

|--🡪 Exit pal\_fn\_voipManagerMsgHandler

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CC \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

pal\_fn\_msghandler()

|--🡪 pal\_internal\_msg\_mgr::getInstance()->process\_msg();

| |--🡪 case Pal\_en\_callControlFsm:

| | |--🡪pal\_fn\_internalDerivedCcEvent //

| | | |--🡪 case Pal\_en\_callControlFsm:

| | | | |-🡪 (\*pal\_glb\_callStateHandlerArray[dNextPendState])( pTermProfile, pCallData, dInputEvent, pErr)

pal\_fn\_ccIdleOnHook

|

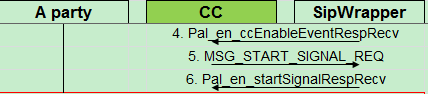
|-🡪 case Pal\_en\_ccOffHook

| |🡪 pal\_fn\_sendQueryHookStatusRequest (pal\_getCcPlatformTxnId(pTermProfile),

| | |-🡪 pal\_fn\_voipSendRequestToPlatform(&dVoipMsg,pErr);

| |🡪 pal\_fn\_setCallStateInfo() // Changing Call state from [CC\_IDLE\_ONHOOK] to [CC\_WAIT\_FOR\_PLAT\_RESP] [CC\_IDLE\_ONHOOK] for Event:[Pal\_en\_ccOffHook

### 2，4~6



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter Sip Wrapper\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipWrapper\_main

|

|🡪 case MSG\_QUERY\_HOOKSTATUS\_REQ:

|🡪 handle\_query\_hookstatus\_req\_from\_ppu()

| |-🡪 p\_resp->msg\_id = (msg\_id\_et)MSG\_QUERY\_HOOKSTATUS\_RESP;

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CC \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Receive: Pal\_en\_ccEnableEventRespRecv from Sip Wrapper

Send: MSG\_START\_SIGNAL\_REQ to SipWrapper

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

pal\_fn\_voipManagerMsgHandler

|🡪case MSG\_QUERY\_HOOKSTATUS\_RESP:

| |--🡪 (\*pal\_glb\_callStateHandlerArray[dCurrentCcState])( (pTermProfile, g\_pCallData, Pal\_en\_ccEnableEventRespRecv, &dErr))

| | |-🡪

Pal\_ty\_retVal pal\_fn\_ccWaitForPlatResponse()

| |--🡪 case Pal\_en\_ccEnableEventRespRecv:

| | |-🡪 \*pal\_glb\_callStateHandlerArray[dNextState])(pTermProfile,pCallData,Pal\_en\_ccEnableEventRespRecv,pErr);

Pal\_ty\_retVal pal\_fn\_ccIdleOnHook

|

|--🡪 case Pal\_en\_ccEnableEventRespRecv:

| |--🡪pal\_fn\_sendStartSignalBasic(pal\_getCcPlatformTxnId(pTermProfile), pTermProfile->dTermId, item, MG\_TIMEOUT, duration, pErr);

| | |-🡪pal\_fn\_sendStartSignalDirection(txn\_id, ICF\_STREAM\_ID\_INVALID,scn\_term\_id, item\_id, sig\_type, sig\_duration, MG\_TONE\_DIRECT\_INTERNAL, pErr);

| |--🡪 pal\_fn\_setCallStateInfo

Exit pal\_fn\_ccIdleOnHook

exit pal\_fn\_voipManagerMsgHandler

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter Sip Wrapper\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Receive: MSG\_START\_SIGNAL\_REQ from PAL

Send: MSG\_START\_SIGNAL\_RESP SUCCESS to PAL

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

SipWrapper\_main()

|

|🡪 case MSG\_START\_SIGNAL\_REQ:

**| |🡪** handle\_start\_signal\_req\_from\_ppu

| | | -🡪 sipw\_playTone()

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Receive: MSG\_START\_SIGNAL\_RESP from Sip Wrapper

Send: MSG\_START\_SIGNAL\_RESP SUCCESS to PAL

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

pal\_fn\_voipManagerMsgHandler()

|

|--🡪 case MSG\_START\_SIGNAL\_RESP:

| |-🡪 (\*pal\_glb\_callStateHandlerArray[dCurrentCcState])( (pTermProfile, g\_pCallData, Pal\_en\_ccEnableEventRespRecv, &dErr))

| | |-🡪

Pal\_ty\_retVal pal\_fn\_ccWaitForPlatResponse()

| |--🡪 case Pal\_en\_startSignalRespRecv:

| | |-🡪 \*pal\_glb\_callStateHandlerArray[dNextState])(pTermProfile,pCallData,Pal\_en\_ccEnableEventRespRecv,pErr);

pal\_fn\_ccIdleOnHook

|

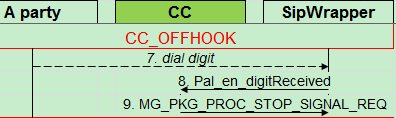
|-🡪 case Pal\_en\_startSignalRespRecv:

| |-🡪pal\_MonitorDM() ///a wrapper over the dl\_start\_monitor\_digitmap

| |-🡪 pal\_fn\_setCallStateInfo()

Exit pal\_fn\_ccIdleOnHook

### 3，7~9



Then terminal dial digit 2, re-enter sipwraper

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SIPWrapper\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipWrapper\_main()

|

|-🡪 handle\_HptMsg(); // handle POTS proxy events and eqpt msg

| |-🡪case GEN\_PULSE\_DIGITAL\_DETECT\_TDM\_SIDE:

| | |-🡪 send\_dtmf\_to\_ppu()

handle\_HptMsg: port 1 digit 2

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_voipManagerMsgHandler

|🡪case MSG\_EVT\_DETECTED\_IND:

| |--🡪 (\*pal\_glb\_callStateHandlerArray[dCurrentCcState])( (pTermProfile, g\_pCallData, Pal\_en\_ccEnableEventRespRecv, &dErr))

| | |-🡪

pal\_fn\_ccOffHook()

|

|-🡪case Pal\_en\_digitReceived:

| |-🡪 pal\_TraverseDM()

| | |-🡪 if (DL\_SUCCESS == dl\_traverse\_digit\_map(transc\_id, digit, p\_ecode))

| | | |-🡪 retcode = dl\_stop\_timer(g\_dl\_trans\_list[transc\_id].timer\_id,transc\_id,p\_ecode);

| | | |-🡪(\*g\_dl\_fdig\_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, digit);

| | | | |-🡪pal\_first\_digit\_evt\_hdlr(txn\_id,service\_id,dl\_register,index,digit)

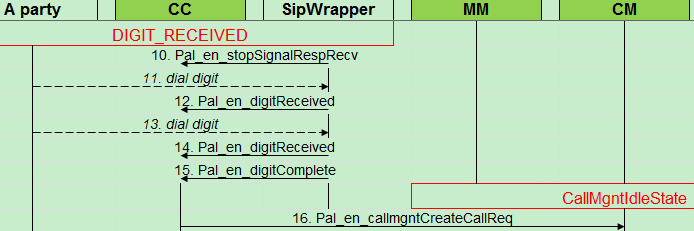
| | | | | |-🡪pal\_fn\_sendStopSignalRequest(0,pTermProfile->dTermId, pTermProfile->dSecondDialTone, &dError);

| | | | | |-🡪pal\_fn\_setCallStateInfo(&((pTermProfile->pCurrentCallContext)->dCallControlState), DIGIT\_RECEIVED,DIGIT\_RECEIVED, Pal\_en\_digitReceived,&dError);

| | | |-🡪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);

### 4，11~16



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_voipManagerMsgHandler

|🡪case MSG\_EVT\_DETECTED\_IND:

| |--🡪 (\*pal\_glb\_callStateHandlerArray[dCurrentCcState])( (pTermProfile, g\_pCallData, Pal\_en\_ccEnableEventRespRecv, &dErr))

| | |-🡪

pal\_fn\_ccOffHook()

|

|-🡪 case Pal\_en\_digitComplete:

| |-🡪 1 subscrible the number to DB

| |-🡪 pal\_fn\_createCallReq

| | |-🡪 buildCallRqstInfo

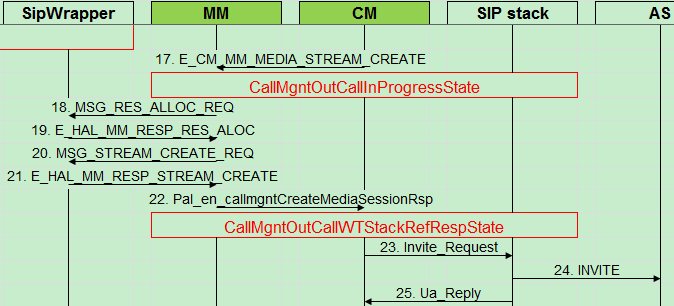
| | | |--🡪 buildCallerInfo

| | | | |🡪 buildLocalUAInfo

| | |-🡪pal\_internal\_msg\_mgr::getInstance()->post\_internal\_message\_on\_term(Pal\_en\_callControlFsm, Pal\_en\_callmgntFsm, Pal\_en\_callmgntCreateCallReq, &pGlbTermProfile[dLineId], (icf\_msg\_st\*)pCreateCallRqst);

| |-🡪pal\_fn\_setCallStateInfo

### 5，17~25



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_internalDerivedCcEvent

|

|-🡪case Pal\_en\_callmgntFsm

| |-🡪 pal\_fn\_callmgntFsmMsgHandler()

| | |-🡪 CallMgntMsgHandler()

| | | send Pal\_en\_callmgntCreateCallReq from CC-🡪 CM

| | | |--🡪 createNewCallReq

| | | | |--🡪 pal\_fn\_checkCallCxtNum

| | | |--🡪 CallMgntHandler::Instance()->DoSendMsgToMM()

| | | | |---🡪 case E\_CM\_MM\_MEDIA\_STREAM\_CREATE:

| | | | | |---🡪 pal\_internal\_msg\_mgr::getInstance()->post\_internal\_message()

pal\_fn\_internalDerivedCcEvent

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter MM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_internal\_msg\_mgr::process\_msg

|

|-🡪 case Pal\_en\_MediaManagerFsm:

| |--🡪 MM\_Handler::handleMsgFromCm(Pal\_st\_internalMsgStrct \*pintlmsg)

| | |--🡪 case E\_CM\_MM\_MEDIA\_STREAM\_CREATE:

| | | |---🡪 handleMediaStreamCreate

| | | | |---🡪 MM\_Handler::getInstance().allocRes(cm\_id, sipw\_stype, rsvRes)

| | | | | |---🡪 pal\_fn\_sendResourceAllocRequest()

| | | | | | |--🡪 case Pal\_en\_audio: ///set Media Type:MG\_SDP\_AUDIO

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SIPW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipWrapper\_main()

|

|--🡪 case case MSG\_RES\_ALLOC\_REQ::

| |--🡪 handle\_res\_alloc\_req\_from\_ppu(userdata,p\_msgHeader);

| | | successful then send MSG\_RES\_ALLOC\_RESP to MM

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter MM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_mgntMsgHandler()

|

|-🡪 case MSG\_RES\_ALLOC\_RESP:

| |-🡪 MM\_Handler::getInstance().handleMsgFromSipw(pMsg);

| | |--🡪 case E\_HAL\_MM\_RESP\_RES\_ALOC:

| | | |--🡪 ret = createStream(cm\_id, reqmsg, port, tmp\_body, err\_resp);

| | | | |--🡪/// create sdp stream

| | | | |---🡪 pal\_fn\_sendStreamCreateOrModifyRequest()

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SIPW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipWrapper\_main()

|

|--🡪 case MSG\_STREAM\_CREATE\_REQ:

| |--🡪 handle\_stream\_create\_req\_from\_ppu()

| | |-🡪 stream\_create\_modify

| | |-🡪 stream\_create\_modify\_voice()

| | | |-🡪 sipw\_voice\_get\_rtp\_payload

|………………………………………..

|

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter MM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_voipManagerMsgHandler

|

|---🡪MM\_Handler::getInstance().handleMsgFromSipw(pMsg);

| |--🡪case E\_HAL\_MM\_RESP\_STREAM\_CREATE:

| | |-🡪pmm\_cxt->DStreamType\_Set();

| | |---🡪 doSendRespToCm(cm\_id,pmsg);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_internalDerivedCcEvent

CallMgntHandler::CallMgntMsgHandler

|

|-🡪case Pal\_en\_callmgntCreateMediaSessionRsp:

| |-🡪CallMgntOutCallInProgressState::createMediaSessionRsp()

| | |--🡪 SendInviteToSsapi(pCallMgntData)

| | | |---🡪 CallMgntHandler::Instance()->DoSendMsgToSSAPIW(pCallMgntData, callmgnt\_ssapi\_invite,body\_in\_msg)

| | | |---🡪pal\_encode\_sdp(\*pCallMgntData->local\_sdp, request.body.sdp);

| | | | | |--🡪 doInviteRequest(pCallMgntData->m\_ua\_request\_id, request, option);

//After DoSendMsgToSSAPIW, the following invite request info will be printed:

allmgnt\_ssapi\_invite request.uac.display\_name.c\_str():["1111"],

request.uac.user\_name.c\_str():[+862111111111], request.uac.random\_contact.c\_str():[+862111111111],request.uac.domain\_name.c\_str():[fs5k8.shanghai.com], request.uac.tag.c\_str():[]request.uac.m\_tag.c\_str():[], request.uac.preferred\_identity.c\_str():[]

callmgnt\_ssapi\_invite request.uas.display\_name.c\_str():["22222222"],request.uas.user\_name.c\_str():[22222222], request.uas.random\_contact.c\_str():[],request.uas.domain\_name.c\_str():[192.168.0.2], request.uas.tag.c\_str():[]request.uas.m\_tag.c\_str():[], request.uas.preferred\_identity.c\_str():[]

callmgnt\_ssapi\_invite request.req\_uri.c\_str():[],

request.alert\_info.c\_str():[], request.call\_id.c\_str():[0000000c-6ee3ffff-000074d-79-3837038@sipua],request.ua\_ref.c\_str():[], request.require.option\_tag.c\_str():[]

callmgnt\_ssapi\_invite request.replaces.totag.c\_str():[],

request.replaces.fromtag.c\_str():[], request.replaces.call\_id.c\_str():[]

callmgnt\_ssapi\_invite request.referredby.scheme=0,

request.referredby.display\_name.c\_str():[], request.referredby.user\_name.c\_str():[]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SSAPI\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

| | |--🡪 ssapi\_parse\_uri(request.req\_uri, pEncInv);

| | |--🡪 SipSSApiInviteRequest\* ss\_inv\_req = pEncInv->EncodeInv(inv, request\_id, request, option);

| | |---🡪 ssapi\_send\_msg(ss\_inv\_req);

| | | |--🡪 sendToSipstack(SipSSApiMsg\* applmsg) // [SAPI]:send to sipstack

| | | | |----🡪 m\_connection.printMsg(applmsg);

| | | | |----🡪 m\_connection.writeMsg(buffer, YAPI\_FALSE);

The invite request info as following:

[SAPI 07/03/70 00:21:21.990522 ]Invite\_Request (103) ; len = 888

nbr of attribs = 8

Application\_Reference: 31 32 2d 30

Alert\_Timeout: 00 3c

SipMsg: INVITE sip:22222222@192.168.0.2;user=phone SIP/2.0

Route: <sip:192.168.0.2;lr>

From: "1111" <sip:+862111111111@fs5k8.shanghai.com;user=phone>

To: "22222222" <sip:22222222@192.[SAPI 07/03/70 00:21:21.990562 ]168.0.2;user=phone>

Call-ID: 0000000c-6ee3ffff-000074d-79-3837038@sipua

CSeq: 1 INVITE

Contact: "1111" <sip:+862111111111@192.168.0.110:50177>

Accept: application/sdp

Allow: INVITE,ACK,CANCEL,BYE,PRACK,UPDATE,INFO,NOTIFY,OPTIONS,REFER,SUBSCRIBE

Supported: timer

Allow-Events: refer

P-Preferre[SAPI 07/03/70 00:21:21.990642 ]d-Identity: sip:+862111111111@fs5k8.shanghai.com

Content-Type: application/sdp

Content-Length: 232

P-Early-Media: supported

v=0

o=ICF 1 0 IN IP4 192.168.0.110

s=Session

c=IN IP4 192.168.0.110

t=0 0

m=audio 50178 RTP/AVP 8 0 18 13

a=rtpmap:8 PCMA/8000/1

a=ptime:20

a=rtpmap:0 PCMU/8000/1

[SAPI 07/03/70 00:21:21.990682 ]a=rtpmap:18 G729/8000/1

a=rtpmap:13 CN/8000/1

a=sendrecv

100Rel: 00

Session\_Expires: 00 00 00 00

UpdateAlertingTO: 01

DNSPool\_Id: 00 01

SessionRefresher: 00 00

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SSAPI recv from sipstack \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

onRecvMsg()

|

|--🡪 printMsg(applmsg);

|--🡪 m\_client->sendToApplication(applmsg);

====the recv from sip stack as following:

Ua\_Reply (104) ; len = 36

nbr of attribs = 3

Application\_Reference: 31 32 2d 30

Ua\_Ref: 30 30 38 32 2d 30 30 30 30 30

30 30 39 2d 30 33 65 34

Dialog\_Ref: 00 00

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipDispatcher::onUaReply()

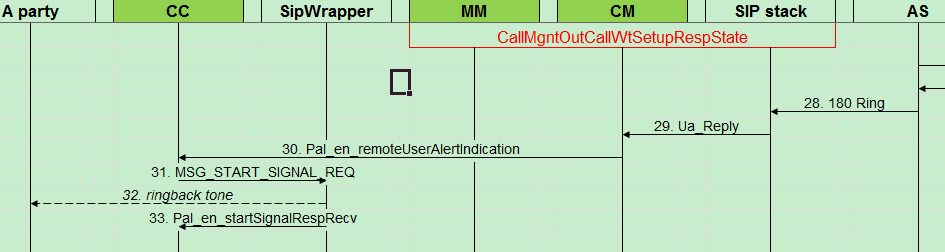
|

|--🡪callmgntH->OnUaReply(request\_id, ua\_ref, reply);

| |-🡪ret = callmgntData->m\_callmgnt\_state->InviteReplyWith0(callmgntData, ua\_ref, reply);

| | WAIT INVITE ACK STATE: InviteReplyWith0 for Terminal: 1

### 6，26~33



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SSAPI recv from sipstack \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

onRecvMsg()

|

|--🡪 printMsg(applmsg);

|--🡪 m\_client->sendToApplication(applmsg);

====the recv from sip stack as following:

[SAPI 06/01/70 00:00:41.015977 ]Ua\_Reply (104) ; len = 823

nbr of attribs = 6

Application\_Reference: 36 2d 30

Ua\_Ref: 30 30 38 32 2d 30 30 30 30 30

30 30 33 2d 30 36 38 30

Dialog\_Ref: 00 00

Method: 00 01

SipMsg: SIP/2.0 180 Ringing

From: <sip:+13288615702@qd.lucentlab.com;user=phone>;tag=0082-00000003-0680

To: <sip:8[SAPI 06/01/70 00:00:41.016058 ]8615003@qd.lucentlab.com;user=phone>;tag=54a262c8-552cbbdb3473ec4c-gm-po-lucentPCSF-081753

Call-ID: 00000006-50d1ffff-00015c3-7904-6760920@sipua

CSeq: 1 INVITE

Contact: <sip:lucentNGFS-094565@135.252.37.124:5060>

Allow: INVITE,BYE,REGISTER,ACK,OPTIONS,CANCEL,SUBSCRIBE,NOTIFY,INFO,REFER,UPDATE

S[SAPI 06/01/70 00:00:41.016122 ]erver: Alcatel-Lucent-HPSS/3.0.3

P-Asserted-Identity: <sip:+13288615003@qd.lucentlab.com;user=phone>

P-Access-Network-Info: xDSL;dsl-location=";21.1.1.73;13288615003;;;"

Content-Length: 0

Session-ID: cc35e9806a2f21763564363662343636

HeaderOffsets: 00 03 00 1b 00 5f 00 04 00 66

00 c5 00 05[SAPI 06/01/70 00:00:41.016179 ] 00 d1 00 fc 00 06

01 05 01 0c 00 09 01 18 01 42

00 12 01 4c 01 51 00 12 01 53

01 55 00 12 01 57 01 5e 00 12

01 60 01 62 00 12 01 64 01 6a

00 12 01 6c 01 71 00 12 01 73

01 7b 00 12 01 7d 01 82 00 12

01 84 01 87 00 12 01 89 01 8d

00 12 01 8f 01 94 00 22 01 9f

01 b7 00 38 01 cf 01 fc[SAPI 06/01/70 00:00:41.016229 ] 00 4a

02 16 02 42 00 54 02 55 02 55

00 60 02 64 02 83

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipDispatcher::onUaReply()

|

|--🡪callmgntH->OnUaReply(request\_id, ua\_ref, reply);

| |-🡪 DoSendMsgToPMC(termidx, msgId, isSupported);

| |-🡪ret = callmgntData->m\_callmgnt\_state->InviteReplyWith1xx(callmgntData, ua\_ref, reply);

| | |-🡪 pDialog = CallMgntHandler::Instance()->MultiEarlyDialogsGetActiveDialog(pCallMgntData, ua\_ref, reply);

| | |-🡪 CallMgntHandler::Instance()->MultiEarlyDialogsFor18xHandler(pDialog, pCallMgntData, ua\_ref, reply);

| | | |-🡪 MuteEarlyMediaAndPlayLocalTone

| | | | |-🡪DoSendMsgToCallHandler (pCallMgntData, Pal\_en\_requireIndialogInd);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_internalDerivedCcEvent

|-🡪 case Pal\_en\_callControlPreHandler:

| |-🡪 pal\_call\_control\_handler(pInternalDerivedEvt);

| | |-🡪 (\*pal\_glb\_callStateHandlerArray[(call\_context->dCallControlState).dCurrentState])();

| | | |-🡪 pal\_fn\_ccCallInitiated(pTermProfile, pCallData, dInputEvent, \*pErr);

| | | | case Pal\_en\_remoteUserAlertIndication:

| | | | pal\_fn\_sendStartSignalBasic (pal\_getCcPlatformTxnId(pTermProfile),pTermProfile->dTermId,MG\_S\_RING\_BACK\_TONE, MG\_TIMEOUT,MG\_S\_RING\_BACK\_TONE\_DURATION,pErr);

| | | | |-🡪 pal\_fn\_sendStartSignalDirection(txn\_id, ICF\_STREAM\_ID\_INVALID, scn\_term\_id, item\_id, sig\_type, sig\_duration, MG\_TONE\_DIRECT\_INTERNAL, pErr);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SIPW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipWrapper\_main()

|

|--🡪 case case MSG\_START\_SIGNAL\_REQ::

| |-🡪 handle\_start\_signal\_req\_from\_ppu(userdata, p\_msgHeader);

| | |-🡪 retCode = sipw\_playTone(termId, p\_ppm\_param->stream\_id, tone, side);

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_voipManagerMsgHandler

|-🡪 case MSG\_START\_SIGNAL\_RESP:

| |-🡪(\*pal\_glb\_callStateHandlerArray[(call\_context->dCallControlState).dCurrentState])();

| | |-🡪pal\_fn\_ccWaitForPlatResponse (pTermProfile, pCallData, dInputEvent, pErr);

| | |-🡪 case Pal\_en\_startSignalRespRecv:

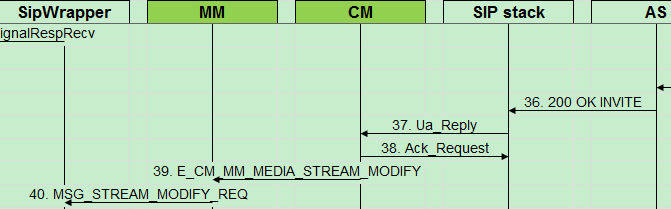
| | | |-🡪 (\*pal\_glb\_callStateHandlerArray[dNextState]) (pTermProfile,pCallData,Pal\_en\_startSignalRespRecv,pErr);

| | | | |-🡪 pal\_fn\_ccCallInitiated(pTermProfile, pCallData, dInputEvent, \*pErr);

| | | | | |-🡪 case Pal\_en\_startSignalRespRecv:

| | | | | | |-🡪 pal\_fn\_setCallStateInfo(&(pCallContext->dCallControlState), BUSYTONE,BUSYTONE,dInputEvent,pErr);

### 7，34~40



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SSAPI recv from sipstack \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

onRecvMsg()

|

|--🡪 printMsg(applmsg);

|--🡪 m\_client->sendToApplication(applmsg);

====the recv from sip stack as following:

[SAPI 06/01/70 00:00:43.320140 ]Ua\_Reply (104) ; len = 1183

nbr of attribs = 6

Application\_Reference: 36 2d 30

Ua\_Ref: 30 30 38 32 2d 30 30 30 30 30

30 30 33 2d 30 36 38 30

Dialog\_Ref: 00 00

Method: 00 01

SipMsg: SIP/2.0 200 OK

From: <sip:+13288615702@qd.lucentlab.com;user=phone>;tag=0082-00000003-0680

To: <sip:88615[SAPI 06/01/70 00:00:43.320212 ]003@qd.lucentlab.com;user=phone>;tag=54a262c8-552cbbdb3473ec4c-gm-po-lucentPCSF-081753

Call-ID: 00000006-50d1ffff-00015c3-7904-6760920@sipua

CSeq: 1 INVITE

Contact: <sip:lucentNGFS-094565@135.252.37.124:5060>

Allow: INVITE,BYE,REGISTER,ACK,OPTIONS,CANCEL,SUBSCRIBE,NOTIFY,INFO,REFER,UPDATE

Serve[SAPI 06/01/70 00:00:43.320271 ]r: Alcatel-Lucent-HPSS/3.0.3

P-Asserted-Identity: <sip:+13288615003@qd.lucentlab.com;user=phone>

P-Access-Network-Info: xDSL;dsl-location=";21.1.1.73;13288615003;;;"

Content-Type: application/sdp

Content-Length: 320

Session-ID: cc35e9806a2f21763564363662343636

v=0

o=LucentPCSF 1627728724 16[SAPI 06/01/70 00:00:43.320326 ]27728724 IN IP4 imsgroup0-000.ism01.qd.lucentlab.com

s=-

c=IN IP4 21.1.1.73

t=0 0

m=audio 50182 RTP/AVP 18 97 96

a=rtpmap:18 G729/8000/1

a=rtpmap:97 PCMA/8000/1

a=rtpmap:96 telephone-event/8000

a=fmtp:96 0-15,16-255

a=sendrecv

a=ptime:20

a=gpmd:97 vbd=yes

a=maxmptime:20 20 -

HeaderOff[SAPI 06/01/70 00:00:43.320383 ]sets: 00 03 00 16 00 5a 00 04 00 61

00 c0 00 05 00 cc 00 f7 00 06

01 00 01 07 00 09 01 13 01 3d

00 12 01 47 01 4c 00 12 01 4e

01 50 00 12 01 52 01 59 00 12

01 5b 01 5d 00 12 01 5f 01 65

00 12 01 67 01 6c 00 12 01 6e

01 76 00 12 01 78 01 7d 00 12

01 7f 01 82 00 12 01 84 01 88

00 12 [SAPI 06/01/70 00:00:43.320436 ]01 8a 01 8f 00 22 01 9a

01 b2 00 38 01 ca 01 f7 00 4a

02 11 02 3d 00 53 02 4e 02 5c

00 54 02 6f 02 71 00 60 02 80

02 9f 62 00 02 a4 03 e3

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipDispatcher::onUaReply()

|

|--🡪callmgntH->OnUaReply(request\_id, ua\_ref, reply);

| |-🡪 ret = callmgntData->m\_callmgnt\_state->InviteReplyWith2xx(callmgntData, ua\_ref, reply);

| | |-🡪 pDialog = CallMgntHandler::Instance()->MultiEarlyDialogsGetActiveDialog(pCallMgntData, ua\_ref, reply);

| | |-🡪 CallMgntHandler::Instance()->DoSendMsgToSSAPIW(pCallMgntData, callmgnt\_ssapi\_ack);

===SSAPI收到ACK后发送给SIP Stack

| | |-🡪 CallMgntHandler::Instance()->MultiEarlyDialogsFor200Handler(pDialog, pCallMgntData, ua\_ref, reply);

| | | |-🡪 CallMgntHandler::Instance()->DoSendMsgToMM(pCallMgntData, E\_CM\_MM\_MEDIA\_STREAM\_MODIFY, VOICE\_STREAM);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter MM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MM\_Handler::handleMsgFromCm(Pal\_st\_internalMsgStrct \*pintlmsg)

|-🡪 case E\_CM\_MM\_MEDIA\_STREAM\_MODIFY:

| |-🡪 rc = modifyStream(cm\_id, nego\_localsdp, \*cm\_reqmsg, send\_to\_sipw, resp);

| | |-🡪 rc = buildMedialines(remotesdp.media\_lines, sipw\_lcalmlines, sipw\_rmtmlines, negomlines, media\_mode, port, session\_mode, PEarlyMedia, v152\_nego\_succ, rule, operation);

| | |-🡪 nego\_localsdp.buildUp(host\_addr, negomlines, sdp\_session\_name, remotesdp.time);

| | |-🡪 pmm\_cxt->V152\_Nego\_Set(v152\_nego\_succ);

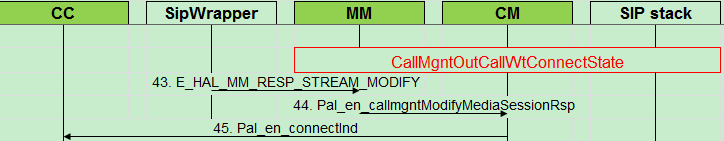
| | |-🡪pmm\_cxt->Nego\_Local\_SDP\_Set(&nego\_localsdp, &remotesdp, cm\_reqmsg.sdpOfferAnswerMode);

| | |-🡪ConvertMedialinesToSipw(sipw\_lcalmlines, &sipw\_lcalstream, &(nego\_localsdp.connection));

| | |-🡪Pal\_en\_fail == pal\_fn\_sendStreamCreateOrModifyRequest(txn\_id, ptermpfl->dTermId, (mg\_api\_id\_et)MSG\_STREAM\_MODIFY\_REQ, dmode,

sipw\_stype, &sipw\_lcalstream, &sipw\_rmtstream, &err,cm\_reqmsg.dialog\_id, cm\_reqmsg.session\_state)

### 8，41~45



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SIPW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipWrapper\_main()

|--🡪 case MSG\_STREAM\_MODIFY\_REQ::

| |-🡪 handle\_stream\_modify\_req\_from\_ppu(userdata, p\_msgHeader);

| | |-🡪 stream\_create\_modify(termId, p\_msg, p\_resp, p\_stream) != noError

| | | |-🡪 rc |= stream\_create\_modify\_voice(termId, p\_msg, p\_stream);

| | | | |-🡪rc |= sipw\_exchangeDsp(termId, p\_stream, voice\_side);

| | | | |-🡪sipw\_voice\_set\_rtp\_transport(termId, p\_stream, p\_msgData);

| | | | | |-🡪rc |= sipw\_configureRemoteAddress(p\_stream->dspRes->connId, p\_stream->sessionHdr.srcPort, local\_IP, ip\_gw, remote\_UDP, remote\_IP,termId);

| | | | |-🡪sipw\_voice\_get\_rtp\_payload(termId, p\_msgData, p\_msg->msg\_id, &conf\_mask, &codec\_type, &rfc2833\_config, &payload\_number);

| | | | |-🡪sipw\_voice\_get\_negotiationChnlParam(termId, p\_stream->sessionAttr.attr.voice.rfc2833Config.dtmf.payloadType, p\_msgData, &negotiationChnlParam, conf\_mask, pkt\_time, codec\_type, rfc2833\_config, payload\_number);

| | | | |-🡪rc |= sipw\_negotiateChannelParam(termId, p\_stream, &negotiationChnlParam,FALSE);

| | | | |-🡪rc |= sipw\_set\_voice\_path(p\_msgData->stream\_mode, p\_stream);

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter MM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_voipManagerMsgHandler

|

|-🡪MM\_Handler::getInstance().handleMsgFromSipw(pMsg);

|-🡪 case E\_HAL\_MM\_RESP\_STREAM\_MODIFY:

| |-🡪 pmm\_cxt->UpdateFSM(pmm\_state);

| |-🡪 pmm\_cxt->DStreamType\_Set();

| |-🡪 doSendRespToCm(cm\_id,pmsg);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_internalDerivedCcEvent

|-🡪 case Pal\_en\_CallManagementDisp:

| |-🡪 pal\_cm\_MediaMsgDispatcher(pInternalDerivedEvt);

| | |-🡪 pal\_fn\_callmgntFsmMsgHandler(term\_profile, pEvent->pCallData, pEvent->dInputEvent.dCallMgntEvent, &dErr);

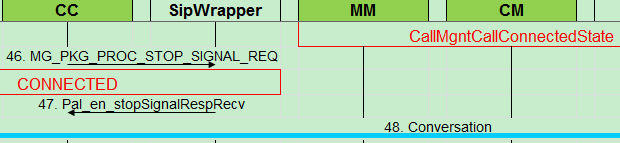
| | | |-🡪callmgntH->CallMgntMsgHandler(pTermProfile->dLineId, pCallData, dInputEvent, pErr);

| | | | |-🡪case Pal\_en\_callmgntModifyMediaSessionRsp:

| | | | | |-🡪result = pcallmgntData->m\_callmgnt\_state->ModifyMediaSessionRsp(pcallmgntData, p\_mm\_resp);

| | | | | | |-🡪CallMgntHandler::Instance()->DoSendMsgToCallHandler(pCallMgntData, Pal\_en\_connectInd);

### 9，46~48



/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_internalDerivedCcEvent

|-🡪 case Pal\_en\_callControlPreHandler:

| |-🡪 pal\_call\_control\_handler(pInternalDerivedEvt);

| | |-🡪 (\*pal\_glb\_callStateHandlerArray[(call\_context->dCallControlState).dCurrentState])();

| | | |-🡪pal\_fn\_ccRingback(pTermProfile,pCallData,dInputEvent,pErr);

| | | | |-🡪case Pal\_en\_connectInd:

| | | | | |-🡪if(Pal\_en\_fail == pal\_fn\_ccCommonInputEventHandle(pTermProfile, pCallData, dInputEvent, pErr))

| | | | | | |-🡪case Pal\_en\_connectInd:

| | | | | | | |-🡪(void)pal\_fn\_sendStopSignalRequest(0,pTermProfile->dTermId, MG\_S\_RING\_BACK\_TONE, pErr);

| | | | | |-🡪pal\_fn\_setCallStateInfo(&(pCallContext->dCallControlState), CONNECTED, CONNECTED, Pal\_en\_connectInd,pErr);

| | |-🡪 cont = handler->HandleMsgHook(event);

| | | |-🡪pal\_setDialogInfoInCallContext(pIcfMsg, call\_context, Pal\_en\_connectInd, pTermProfile, &err);

| | | |-🡪pal\_fn\_startCallStatisticHandle( pTermProfile, call\_context);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter SIPW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SipWrapper\_main()

|--🡪 case case MSG\_STOP\_SIGNAL\_REQ:::

| |-🡪 handle\_stream\_modify\_req\_from\_ppu(userdata, p\_msgHeader);

| | |-🡪 tone = get\_tone\_id(p\_ppm\_param->item\_id);

| | |-🡪case 具体哪个case不清楚

| | | |-🡪retCode = sipw\_playTone(termId, p\_ppm\_param->stream\_id, 0xFF, GEN\_NONE\_SIDE);

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

| | |-🡪sipw\_displayDspStreamByTerm(pTerm);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Enter CC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pal\_fn\_voipManagerMsgHandler

|-🡪 case MSG\_STOP\_SIGNAL\_RESP:

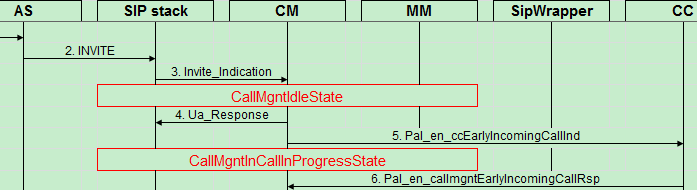
| |-🡪 (\*pal\_glb\_callStateHandlerArray[(pTermProfile->pCurrentCallContext->dCallControlState).dCurrentState]) (pTermProfile, g\_pCallData, Pal\_en\_stopSignalRespRecv, &dErr);

| | |-🡪 pal\_fn\_ccConnected(pTermProfile, pCallData, dInputEvent, pErr);

| | | |-🡪 case Pal\_en\_stopSignalRespRecv:

## 2，被叫端流程

### 1，1~6



**/======================enter ssapi\_wrapper=====================================**

pal\_fn\_msghandler()

|

|-🡪case ICF\_MODULE\_ICF:

| |-🡪ssapi\_parse\_msg()

| | |-🡪case SipSSApiMsg::E\_INVITE\_INDICATION:

| | | |--🡪ssapi\_parse\_invite\_indication()

| | | |-🡪SipEngine::getInstance().getCb()->onInviteIndication( ind\_ref, InviteInd);

**/======================enter CM=====================================**

CallMgntHandler::onInviteIndication(const SSAPIW::UaRef& ua\_ref, const SSAPIW::InviteIndication& InviteInd)

|-🡪CallMgntHandler::Instance()->DoSendMsgToPMC(pGlbTermProfile[line\_id].dTermId, MSG\_PMC\_ANSWERED\_INCOMING\_CALL\_NOTIFY, Pal\_en\_true);

|-🡪 ret = callmgntData->m\_callmgnt\_state->newCallIncoming(callmgntData, ua\_ref, InviteInd);

| |--🡪Pal\_en\_success != formDisplayInf()

| |--🡪 get incoming number

| |--🡪CallMgntHandler::DoSendMsgToSSAPIW

|-🡪Exit CallMgntHandler::onInviteIndication.

**/======================enter CC=====================================**

Pal\_ty\_void pal\_internal\_msg\_mgr::process\_msg()

|

|-🡪pal\_fn\_internalDerivedCcEvent(&internalDerivedEvt);

|---|-🡪 case case Pal\_en\_callControlPreHandler:

| | |-🡪pal\_call\_control\_handler(pInternalDerivedEvt);

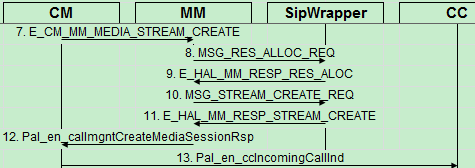
| | | |-🡪CallControlMsg<Pal\_en\_ccEarlyIncomingCallInd,Pal\_en\_preAct>::HandleMsgHook (Pal\_st\_internalMsgStrct\* event)

| | | |…….

| | | |---🡪 pal\_fn\_sendEarlyIncCallCfm

| |　| | |--🡪al\_internal\_msg\_mgr::getInstance()->post\_internal\_message\_on\_term

### 2，7~13



**/======================enter CM=====================================**

pal\_fn\_internalDerivedCcEvent

|

|-🡪case Pal\_en\_callmgntFsm:

| |-🡪pal\_fn\_callmgntFsmMsgHandler

| | |🡪callmgntH->CallMgntMsgHandler(pTermProfile->dLineId, pCallData, dInputEvent, pErr);

| | | |-🡪 case Pal\_en\_callmgntEarlyIncomingCallRsp:

| | | | |-🡪 pcallmgntData->m\_callmgnt\_state->earlyIncomingCallRsp(pcallmgntData, early\_incall\_resp);

| | | | --🡪 DoSendMsgToMM //send MSG\_RES\_ALLOC\_REQ to sipwrapper

**/======================enter MM=====================================**

pal\_mm\_internalMsgDispatcher:

handleMediaStreamCreate

|

|-🡪allocRes ///

**/======================enter SIPW=====================================**

SipWrapper\_main

|

|-🡪case MSG\_RES\_ALLOC\_REQ:

| |-🡪handle\_res\_alloc\_req\_from\_ppu

| | /\*alloc dsp resource \*/

| |/\* return MSG\_RES\_ALLOC\_RESP to PAL\*/

**/======================enter MM=====================================**

MM\_Handler::handleMsgFromSipw

|

|---🡪Case E\_HAL\_MM\_RESP\_RES\_ALOC

| |-🡪 createStream() ///send MSG\_STREAM\_CREATE\_REQ

**/======================enter SIPW=====================================**

SipWrapper\_main

|

|-🡪case MSG\_STREAM\_CREATE\_REQ:

| |-🡪handle\_stream\_create\_req\_from\_ppu

| | /\*alloc dsp resource \*/

| |/\* return MSG\_STREAM\_CREATE\_RESP

**/======================enter MM=====================================**

pal\_fn\_msghandler

|

|-🡪pal\_fn\_voipManagerMsgHandler

| |-🡪 case MSG\_STREAM\_CREATE\_RESP:

| | |-🡪MM\_Handler::getInstance().handleMsgFromSipw(pMsg)

| | | |--🡪 doSendRespToCm(cm\_id,pmsg);

**/======================enter CM=====================================**

pal\_fn\_internalDerivedCcEvent // used send msg from CM to CC

|

|----case case Pal\_en\_CallManagementDisp:

| |-🡪 pal\_cm\_MediaMsgDispatcher

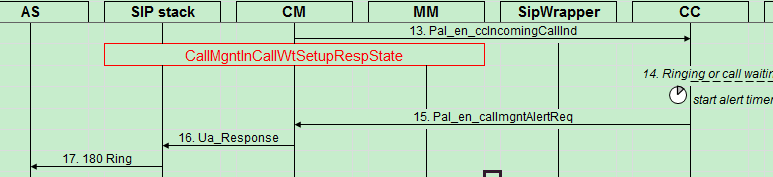
| | 　|--🡪 pal\_fn\_callmgntFsmMsgHandler

| | | |---🡪 callmgntH->CallMgntMsgHandler(pTermProfile->dLineId, pCallData, dInputEvent, pErr);

| | | | |-🡪 case Pal\_en\_callmgntCreateMediaSessionRsp:

| | | | | |-🡪 DoSendMsgToCallHandler

### 3，14~18



**/======================enter CC=====================================**

pal\_call\_control\_handler

|--🡪 Pal\_en\_ccIncomingCallInd

pal\_fn\_ccIdleOnHook

|

|-🡪 case Pal\_en\_ccIncomingCallInd:

| |-🡪 Pal\_en\_fail==pal\_fn\_idleOnHook\_IncomingCallInd(pTermProfile,…

| | |-🡪/\* /\* Play ring tone and display caller id\*/ \*/

| | |-🡪 pal\_fn\_sendStartSignalRingingCallId

| |--🡪/\* pal\_fn\_idleOnHook\_IncomingCallInd: play ringing with callid, cadance is 1\*/

| |-🡪exit pal\_fn\_idleOnHook\_IncomingCallInd.

**/======================enter SIPW=====================================**

SipWrapper\_main

|

|-🡪 case MSG\_START\_SIGNAL\_REQ:

| |-🡪 handle\_start\_signal\_req\_from\_ppu

| | |🡪 sipw\_getToneSide

| | |--🡪 get\_tone\_id

| | |--🡪 send MSG\_START\_SIGNAL\_RESP SUCCESS to PAL

| | |-🡪

**/======================enter CC=====================================**

Changing Call state from [CC\_IDLE\_ONHOOK] to [RINGING]

pal\_fn\_voipManagerMsgHandler:

|

|--🡪 case MSG\_START\_SIGNAL\_RESP:

| pal\_fn\_voipManagerMsgHandler

| |--🡪 Pal\_en\_startSignalRespRecv

pal\_fn\_ccIdleOnHook

|

|--🡪 pal\_fn\_sendUserAlertedReq:

| |-🡪 buildAlertRqstInfo

| |-🡪 pal\_internal\_msg\_mgr::getInstance()->post\_internal\_message\_on\_term //Send to CM

**/======================enter CM=====================================**

CallMgntHandler::CallMgntMsgHandler

|-🡪 case Pal\_en\_callmgntAlertReq:

| |--🡪 pcallmgntData->m\_callmgnt\_state->sendAlertReq(pcallmgntData, alert\_req)

| | |--🡪 CallMgntInCallWtSetupRespState::sendAlertReq

| | | |--🡪 CallMgntHandler::Instance()->DoSendMsgToSSAPIW(pCallMgntData

/\*send to sipstack\*\*/

180 methord 1 for Terminal:4 in state 9,voiceservertatus=1,lineId=3Contact header params: display\_name="2222", user\_name=+862122222222, random\_contact=+862122222222,

The info sent to sipstack as following:

a\_Response (107) ; len = 250

nbr of attribs = 7

Application\_Reference: 31 33 2d 30

Ua\_Ref: 30 30 38 32 2d 30 30 30 30 30

30 30 61 2d 30 33 65 35

Dialog\_Ref: 00 00

Method: 00 01

SipMsg: SIP/2.0 180

Contact: "2222" <sip:+862122222222@192.168.0.110:50177>

Allow: INVITE,ACK,CANCEL,BYE,PR[SAPI 07/03/70 00:21:22.172322 ]ACK,UPDATE,INFO,NOTIFY,OPTIONS,REFER,SUBSCRIBE

Allow-Events: refer

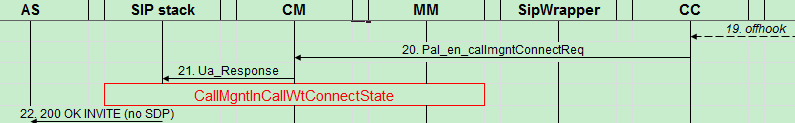
Content-Length: 0

DNSPool\_Id: 00 01

SessionRefresher: 00 00

4，

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****



*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**/======================enter SIPW=====================================**

SipWrapper\_main

handle\_HptMsg

|

|--🡪case GEN\_HOOK\_STATE\_CHANGED:

| |--🡪 handle\_term\_ev\_offhook

| | |--🡪 send\_event\_to\_ppu

**/======================enter CC=====================================**

pal\_fn\_voipManagerMsgHandler

|

|--🡪 Current Line Feeding is parked, Req Line Feeding is normal

|--🡪 Send MSG\_LINE\_FEEDING\_REQ to Plat, polarType:normal

|--🡪 pal\_fn\_ccRinging

|--🡪|-🡪 case Pal\_en\_ccOffHook:

| | |-🡪 pal\_fn\_sendConnectReq // Sending MG\_PKG\_PROC\_STOP\_SIGNAL\_REQ(50) for 4

| | | |-🡪 pal\_fn\_setCallStateInfo

**/======================enter SIPW=====================================**

handle\_line\_feeding\_from\_ppu

|

|---🡪 receive MSG\_STOP\_SIGNAL\_REQ from PAL

**/======================enter CC=====================================**

pal\_fn\_ccRinging

|-🡪case Pal\_en\_stopSignalRespRecv:

| |--🡪Pal\_en\_stopSignalRespRecv

| | |--🡪pal\_fn\_sendConnectReq

| | | |--🡪buildConnectRqstInfo

**| | | | |--🡪buildLocalUAInfo**

**| | |---🡪pal\_fn\_setCallStateInfo:**

| |--🡪 Sending Connect Request for Incoming Call.

**/======================enter CM=====================================**

CallMgntHandler::CallMgntMsgHandler

|

|--🡪DoSendMsgToSSAPIW

**/======================enter SSAPI=====================================**

receive MSG\_PMC\_RETRIEVE\_CALL\_STATIS\_INFO\_REQ from PAL

the following info is sent to sip stack:

[SAPI 07/03/70 00:21:26.647602 ]Ua\_Response (107) ; len = 461

nbr of attribs = 7

Application\_Reference: 31 33 2d 30

Ua\_Ref: 30 30 38 32 2d 30 30 30 30 30

30 30 61 2d 30 33 65 35

Dialog\_Ref: 00 00

Method: 00 01

SipMsg: SIP/2.0 200

Contact: "2222" <sip:+862122222222@192.168.0.110:50177>

Allow: INVITE,ACK,CANCEL,BYE,PR[SAPI 07/03/70 00:21:26.647682 ]ACK,UPDATE,INFO,NOTIFY,OPTIONS,REFER,SUBSCRIBE

Allow-Events: refer

Content-Type: application/sdp

Content-Length: 178

v=0

o=ICF 1 0 IN IP4 192.168.0.110

s=Session

c=IN IP4 192.168.0.110

t=0 0

m=audio 50184 RTP/AVP 8 13

a=rtpmap:8 PCMA/8000/1

a=ptime:20

a=rtpmap:13 CN/8000/1

a=sendrecv