

当生存能力服务是被实施的时，排除呼叫失败故障

目录表

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[问题](#)

[程序](#)

[解决方案](#)

[简介](#)

本文描述如何排除呼叫失败故障，当Customer Voice Portal (CVP)时生存能力脚本在入口网关的流入的拨号对等体被配置。

贡献用Kabeer Noorudeen，Cisco TAC工程师。

Prerequisites

Requirements

Cisco 建议您了解以下主题：

- CVP全面的呼叫流
- IOS网关和PRI协议
- Cisco Unified智能联络管理(ICM)，Cisco Unified Contact Center Enterprise (UCCE)配置

Components Used

本文档中的信息基于以下软件版本：

- CVP服务器9.0以上

- UCCE 9.0以上
- IOS网关15.X

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

问题

当服务被添加到一流入PoTs Dialpeer触发生存能力脚本时，呼叫失效与此原因：

ccCallDisconnect : 原因Value=81

Note:当呼叫发生故障时，请运行debug命令，**debug voip ccapi inout**，为了发现报告的错误代码。另外，请运行debug命令**调试ccsip消息**，如果要发现报告的SIP错误信息，当呼叫发生故障时。

当**调试ccsip消息**是启用的时，不一致的错误信息报告。SIP消息，不给予征兆除呼叫发生故障的那之外，在网络VRU标签返回后，可以是令人误解的。

如果服务从被取消从流入pots拨号点，一切良好工作。

程序

Step 1.一入口网关enable (event)调试：**debug voip ccapi inout**、**调试ccsip消息**、**调试语音应用全部**和**debug isdn q931**。

步骤2.再现问题。发出呼叫到网关。

步骤3.收集日志。请使用命令，**show logging**。

如果查看CCAPI日志(**debug voip ccapi inout**)，您看到呼叫获得顺利地递交生存能力脚本呼叫然后发生故障：

```

*Sep 13 19:29:57.507 CT : //21/9E2AF1DC800C/CCAPI/cc_process_call_setup_ind :
>>>>CCAPI递交了与标记101的cid 21对app “_ManagedAppProcess_cvp-survivability”
*Sep 13 19:29:58.507 CT : //21/9E2AF1DC800C/CCAPI/ccCallSetupAck :
  呼叫Id=21

*Sep 13 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/cc_api_call_disconnected :
  原因Value=81 , Interface=0x2500E10 , 呼叫Id=21
*Sep 13 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/cc_api_call_disconnected :
  引入呼叫(Responded=FALSE , 原因Value=81 , 重试次数Count=0)
*Sep 13 19:29:58.551 CT : //22/9E2AF1DC800C/CCAPI/ccCallDisconnect :
  原因Value=81 , Tag=0x0 , 引入呼叫(早先断开Cause=0 , 断开Cause=0)
*Sep 13 19:29:58.551 CT : //22/9E2AF1DC800C/CCAPI/ccCallDisconnect :
  原因Value=81 , 引入呼叫(Responded=FALSE , 原因Value=81)
*Sep 13 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/ccCallDisconnect :
  原因Value=81 , Tag=0x0 , 引入呼叫(早先断开Cause=0 , 断开Cause=81)
*Sep 13 19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/ccCallDisconnect :
  原因Value=81 , 引入呼叫(Responded=TRUE , 原因Value=81)

*Sep 13 19:29:57.507 CT : //21/9E2AF1DC800C/CCAPI/cc_process_call_setup_ind
: >>>>CCAPI递交了与标记101的cid 21对app “_ManagedAppProcess_cvp-survivability ” *Sep
13 19:29:58.507 CT : //21/9E2AF1DC800C/CCAPI/ccCallSetupAck : 呼叫Id=21*Sep 13
19:29:58.551 CT : //21/9E2AF1DC800C/CCAPI/cc_api_call_disconnected : 原因Value=81 ,
Interface=0x2500E10 , 呼叫Id=21*Sep 13 19:29:58.551 CT
: //21/9E2AF1DC800C/CCAPI/cc_api_call_disconnected : 引入呼叫(Responded=FALSE , 原因
Value=81 , 重试次数Count=0)*Sep 13 19:29:58.551 CT
: //22/9E2AF1DC800C/CCAPI/ccCallDisconnect : 原因Value=81 , Tag=0x0 , 引入呼叫(早先断
开Cause=0 , 断开Cause=0)*Sep 13 19:29:58.551 CT
: //22/9E2AF1DC800C/CCAPI/ccCallDisconnect : 原因Value=81 , 引入呼叫
(Responded=FALSE , 原因Value=81)*Sep 13 19:29:58.551 CT
: //21/9E2AF1DC800C/CCAPI/ccCallDisconnect : 原因Value=81 , Tag=0x0 , 引入呼叫(早先断
开Cause=0 , 断开Cause=81)*Sep 13 19:29:58.551 CT
: //21/9E2AF1DC800C/CCAPI/ccCallDisconnect : 原因Value=81 , 引入呼叫
(Responded=TRUE , 原因Value=81)

```

如果查看调试为tcl (即debugvoice应用程序全部), 您看到呼叫人被断开, 当tcl设法到当前呼叫状态时。

掠夺者

```

*Sep 13 20:29:54.211 CT : //81//TCL : /tcl_InfotagObjCmd : infotag获得evt_state_current
*Sep 13 20:29:54.211 CT : //81//TCL : /tcl_InfotagGetObjCmd : infotag获得evt_state_current
*Sep 13 20:29:54.211 CT : //81//AFW_:/vtr_ev_state_current : argc 2
*Sep 13 20:29:54.211 CT : //81//AFW_:/vtr_ev_state_current : 事件[CALL_INIT]
*Sep 13 20:29:54.211 CT : //81//TCL : /tcl_FSMObjCmd : fsm setstate
CALLER_DISCONNECTED
*Sep 13 20:29:54.211 CT : //81//TCL : /tcl_FSMSetStateObjCmd : setstate setstate
CALLER_DISCONNECTED
*Sep 13 20:29:54.211 CT : //81//TCL : /tcl_InfotagObjCmd : infotag获得con_ofleg 81
*Sep 13 20:29:54.211 CT : //81//TCL : /tcl_InfotagGetObjCmd : infotag获得con_ofleg 81
*Sep 13 20:29:54.211 CT : //81//AFW_:/vtr_co_ofleg : argc 3 argindex 2

```

*Sep 13 20:29:54.211 CT : //81//Tcl : /tcl_parseCallID_vartagObj : VARTAG转换段Count=1
*Sep 13 20:29:54.211 CT : //81//AFW_:/vtr_co_ofleg : EV_CONNECTIONS []
*Sep 13 20:29:54.211 CT : //81//TCL : /tcl_LegObjCmd : 段断开81
*Sep 13 20:29:54.211 CT : //81//TCL : /tcl_LegDisconnectObjCmd : 断开81
*Sep 13 20:29:54.211 CT : //81//Tcl : /tcl_parseCallID_vartagObj : VARTAG转换段Count=1
*Sep 13 20:29:54.211 CT : //81//TCL : /tcl_InfotagObjCmd : infotag获得evt_state_current *Sep 13 20:29:54.211 CT : //81//TCL : /tcl_InfotagGetObjCmd : infotag获得evt_state_current *Sep 13 20:29:54.211 CT : //81//AFW_:/vtr_ev_state_current : argc 2 *Sep 13 20:29:54.211 CT : //81//AFW_:/vtr_ev_state_current : 事件[CALL_INIT] *Sep 13 20:29:54.211 CT : //81//TCL : /tcl_FSMObjCmd : fsm setstate CALLER_DISCONNECTED *Sep 13 20:29:54.211 CT : //81//TCL : /tcl_FSMSetStateObjCmd : setstate setstate CALLER_DISCONNECTED *Sep 13 20:29:54.211 CT : //81//TCL : /tcl_InfotagObjCmd : infotag获得con_ofleg 81 *Sep 13 20:29:54.211 CT : //81//TCL : /tcl_InfotagGetObjCmd : infotag获得con_ofleg 81 *Sep 13 20:29:54.211 CT : //81//AFW_:/vtr_co_ofleg : argc 3 argindex 2 *Sep 13 20:29:54.211 CT : //81//Tcl : /tcl_parseCallID_vartagObj : VARTAG转换段Count=1 *Sep 13 20:29:54.211 CT : //81//AFW_:/vtr_co_ofleg : EV_CONNECTIONS [] *Sep 13 20:29:54.211 CT : //81//TCL : /tcl_LegObjCmd : 段断开81 *Sep 13 20:29:54.211 CT : //81//TCL : /tcl_LegDisconnectObjCmd : 断开81 *Sep 13 20:29:54.211 CT : //81//Tcl : /tcl_parseCallID_vartagObj : VARTAG转换段Count=1

现在，如果潜水下来到isdn消息(debug isdn q931)您看到消息不同步与PSTN消息。有进来在设置之后和在呼叫进行前的设备消息。生存能力脚本不能处理此情况。

掠夺者

*Sep 13 19:53:31.763 CT : ISDN Se0/0/0:23 Q931 : RX < -设置pd = 8个callref = 0x1114
*Sep 13 19:53:31.767 CT : ISDN Se0/0/0:23 Q931 : RX < -设备pd = 8个callref = 0x1114
*Sep 13 19:53:31.767 CT : ISDN Se0/0/0:23 **错误** : L3_BadPeerMsg : 事件0x62哥斯达黎加0x9114 callid 0x0
*Sep 13 19:53:31.767 CT : ISDN Se0/0/0:23 Q931 : TX- > RELEASE_COMP pd = 8个callref = 0x9114
*Sep 13 19:53:32.767 CT : ISDN Se0/0/0:23 Q931 : TX- > CALL_PROC pd = 8个callref = 0x9114
*Sep 13 19:53:32.767 CT : ISDN Se0/0/0:23 Q931 : TX- >连接pd = 8个callref = 0x9114
*Sep 13 19:53:32.803 CT : ISDN Se0/0/0:23 Q931 : RX < - RELEASE_COMP pd = 8个callref = 0x1114
*Sep 13 19:53:32.807 CT : ISDN Se0/0/0:23 Q931 : RX < - RELEASE_COMP pd = 8个callref = 0x1114
*Sep 13 19:53:32.807 CT : ISDN Se0/0/0:23 **错误** : L3_BadPeerMsg : 事件0x5A哥斯达黎加0x9114 callid 0x0

*Sep 13 19:53:31.763 CT : ISDN Se0/0/0:23 Q931 : RX < -设置pd = 8 callref = 0x1114 *Sep 13 19:53:31.767 CT : ISDN Se0/0/0:23 Q931 : RX < -设备pd = 8 callref = 0x1114 *Sep 13 19:53:31.767 CT : ISDN Se0/0/0:23 **错误** : L3_BadPeerMsg : 事件0x62哥斯达黎加0x9114 callid 0x0 *Sep 13 19:53:31.767 CT : ISDN Se0/0/0:23 Q931 : TX- > RELEASE_COMP pd = 8 callref = 0x9114 *Sep 13 19:53:32.767 CT : ISDN Se0/0/0:23 Q931 : TX- > CALL_PROC pd = 8 callref = 0x9114 *Sep 13 19:53:32.767 CT : ISDN Se0/0/0:23 Q931 : TX- >连接pd = 8个callref = 0x9114 *Sep 13 19:53:32.803 CT : ISDN Se0/0/0:23 Q931 : RX < - RELEASE_COMP pd = 8 callref = 0x1114 *Sep 13 19:53:32.807 CT : ISDN Se0/0/0:23 Q931 : RX < - RELEASE_COMP

pd = 8 callref = 0x1114 *Sep 13 19:53:32.807 CT : ISDN Se0/0/0:23 **错误** : L3_BadPeerMsg
: 事件0x5A哥斯达黎加0x9114 callid 0x0

解决方案

L3_BadPeerMsg在这里解决此问题的键。当有在交换机类型的不匹配在网关和PSTN之间时，此错误报告。

推荐是运行命令设置PSTN和网关同意的ISDN交换机类型。在此方案中在PSTN的交换机类型是**primary-ni**。对于美国用户，常用的交换类型是**primary-ni**。

在入口网关配置的isdn switch-type命令**primary-ni**解决了问题。