

排除语音连接中继线故障

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Conventions](#)

[问题](#)

[解决方案](#)

[连接trunk的常见问题](#)

[开始排除故障](#)

[确定什么呼叫是UP](#)

[DTMF排除故障](#)

[Related Information](#)

[Introduction](#)

语音连接中继线永久建立语音呼叫，VoIP、帧中继语音(VoFR)或者ATM语音(VoATM)。呼叫建立，当路由器出现，并且配置完成。当语音端口出现，语音端口自动地拨打假的电话号码指定在语音端口下并且发出呼叫到位置。语音端口完成呼叫对另一个末端通过对应的Dial Peer。一旦此连接被建立，就路由器而言，语音呼叫在会话上和被连接。

[Prerequisites](#)

[Requirements](#)

本文档没有任何特定的前提条件。

[Components Used](#)

This document is not restricted to specific software and hardware versions.

本文档中的信息都是基于特定实验室环境中的设备创建的。All of the devices used in this document started with a cleared (default) configuration.如果您的网络实际，请切记您了解所有命令的潜在影响，在您使用它前。

[Conventions](#)

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

问题

适合于对Trunk的常见问题是透明对路由器和非常难排除故障。在语音Trunk看到的常见问题表明，当呼叫在Trunk时发出，并且什么都听不到。这是其中一连接trunk的已知的问题和是由许多不同的问题造成的。另一议题是没有适当地通过的双音多频语音，并且从专用的交换分机(PBX)的信令对PBX没有适当地被传输。本文排除这些问题故障。

当语音卡车是上和活跃的时，信号在连接卡车不同运行。任何命令您通常发出在信号特征的语音端口下不是相关和有用的。语音中继成为信令conduit并且传递在VoIP链路间的信号。当您使用语音Trunk时，pbx信令必须匹配端到端。就两台PBX机器而言，目标是使语音中继连接看起来相同与一条被租用的T1线路与PBX，用完全地透明的路由器，当一条清楚的链路建立在两PBX之间在整个进程时。

当Trunk出现时，Trunk成为软件电缆，并且信号类型认为连接器类型。Trunk对使用的信号类型不关心。Trunk仍然出现，即使信号不配比在两端。只要在两端的PBX执行同样信令，Trunk正常运行。

解决方案

采取的方法，当您跟那排除连接trunk问题故障不同时使用交换呼叫。要看到发生了什么确实，在Trunk被验证后，您需要查找到pbx信令。在您继续查看信令前，请验证Trunk上，并且数字式信号处理器(DSP)处理语音数据包。

Note: 您很可能要禁用语音活动检测(VAD)为了排除故障。一旦被验证Trunk正确地作用，您需要查看电话信令为了进一步排除故障。

如果Trunk建立，并且没人设法做呼叫，中继线保活信息反复传送在远程机箱之间。这些Keepalive验证中继线连通性并且传播从端到端的信令信息。要验证这些Keepalive，请发出[debug vpm signal命令](#)。如果有许多Trunk，[debug vpm](#)的输出发出命令，您能对单个端口限制输出，如果您[debug vpm port x命令](#)选项的问题，“x”是正在考虑中的语音端口。当您查看所有端口时，这是发出的[debug vpm signal命令](#)的输出：

```
21:18:12: [3/0:10(11)] send to dsp sig DCBA state 0x0
21:18:12: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:12(13)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:20(21)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:12(13)] send to dsp SIG DCBA state 0x0
21:18:12: [3/0:20(21)] send to dsp SIG DCBA state 0x0
21:18:12: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:18:12: [3/0:3(4)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:9(10)] rcv from dsp SIG DCBA state 0x0
21:18:12: [3/0:3(4)] send to dsp SIG DCBA state 0x0
21:18:13: [3/0:9(10)] send to dsp SIG DCBA state 0x0
21:18:13: [3/0:19(20)] rcv from dsp SIG DCBA state 0x0
```

如果用[debug vpm port x命令](#)限制此，更加容易的调试解释，如此示例所显示，：

```
21:21:08: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:12: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:13: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:17: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:18: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:22: [3/0:0(1)] send to dsp SIG DCBA state 0x0
```

```
21:21:23: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:27: [3/0:0(1)] send to dsp SIG DCBA state 0x0
21:21:28: [3/0:0(1)] rcv from dsp SIG DCBA state 0x0
21:21:32: [3/0:0(1)] send to dsp SIG DCBA state 0x0
```

发送Keepalive并且接受了每五秒。术语“发送了到DSP”，并且“从DSP接受”请是从Cisco IOS观点。用DSP替代PBX使更加可理解。这些是被看到的消息，当没有在Trunk时的活动。保活信息告诉电路的每个末端的路由器Trunk仍然是。当五这些消息连续时丢失，Trunk断开。如果Trunk在网络，经常拍动其中一个原因是。要验证是否发送语音中继Keepalive并且被接受，请发出**debug vpm trunk-sc**命令。此调试不生成任何输出，直到中继线保活丢失。当Keepalive丢失时，这是**debug vpm trunk-sc**命令输出的示例：

```
22:22:38: 3/0:22(23): lost Keepalive
22:22:38: 3/0:22(23): TRUNK_SC state : TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE
22:22:38: 3/0:22(23): trunk_rtc_set_AIS on
22:22:38: 3/0:22(23): trunk_rtc_gen_pattern : SIG pattern 0x0
22:22:38: 3/0:22(23): TRUNK_SC, TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
22:22:39: 3/0:13(14): lost Keepalive
22:22:39: 3/0:13(14): TRUNK_SC state : TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE
22:22:39: 3/0:13(14): trunk_rtc_set_AIS on
22:22:39: 3/0:13(14): trunk_rtc_gen_pattern : SIG pattern 0x0
22:22:39: 3/0:13(14): TRUNK_SC, TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
```

如果输出没有被看到，当发出时**debug vpm trunk-sc**命令，则Keepalive没有丢失。即使Keepalive丢失，Trunk坚持，直到五个连续消息丢失。这意味着连接必须发生故障为25秒，在Trunk断开前。

连接trunk的常见问题

有与语音中继连接产生关联的几个Bug。如果看到异常的任何东西请检查这些Bug。当发布了的时候Cisco IOS Software 12.2，大多这些问题解决了并且集成。您能通过Bug查找意识自己这些是问题的原因更旧的代码的。多数常见问题之一是获得PBX在中继线连接正确地发信号。它似乎类似一个好想法减少Trunk和配置路由器，以便他们工作在每个末端，但是方法是确实妨碍达到预期目的，因为您当前更改的任何变得需讨论Trunk一次设立。排除故障的最佳方法是功能的Trunk上和。

开始排除故障

查看基础设立是必要的这些正确地作用：

- Trunk建立？发出**show voice call summary**命令，并且切记Trunk在S_CONNECTED状态。
- DSP处理是否是信息包？发出**show voice dsp**命令验证此。如果看不到信息包由DSP处理，这是因为VAD是启用的并且是抑制信息包。关闭VAD，重建Trunk并且再查找。并且，请检查信息包计数器增加，当发出时**show call active voice brief**命令。此命令也显示VAD是否为呼叫登录问题是启用的。

如果Trunk连接到模拟端口在任何站点，验证PBX的操作在非中继的模式下是最佳的。要排除模拟E&M连接问题故障，请参见[了解和排除模拟E&M接口类型和布线故障](#)。一旦一切正确验证并且作用，请提出Trunk并且查看被通过在PBX之间的信令。

理想的方式排除语音中继连接问题故障将检查被通过在PBX之间的信令。这是最佳有远程登录会话到正在考虑中每个的路由器，以便信令可以被观察作为它从一端通过到其他。本文使用E&M闪烁信令，因为是相当普遍的，并且闪烁定时必须被考虑到。

这是发起呼叫的路由器的输出被连接到PBX：

```
22:22:38: 3/0:22(23): lost Keepalive
```

```
22:22:38: 3/0:22(23): TRUNK_SC state : TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE
22:22:38: 3/0:22(23): trunk_rtc_set_AIS on
22:22:38: 3/0:22(23): trunk_rtc_gen_pattern : SIG pattern 0x0
22:22:38: 3/0:22(23): TRUNK_SC, TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
22:22:39: 3/0:13(14): lost Keepalive
22:22:39: 3/0:13(14): TRUNK_SC state : TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE
22:22:39: 3/0:13(14): trunk_rtc_set_AIS on
22:22:39: 3/0:13(14): trunk_rtc_gen_pattern : SIG pattern 0x0
22:22:39: 3/0:13(14): TRUNK_SC, TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
```

此输出显示路由器终止呼叫。网络时间协议(NTP)是同步的。

```
22:22:38: 3/0:22(23): lost Keepalive
22:22:38: 3/0:22(23): TRUNK_SC state : TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE
22:22:38: 3/0:22(23): trunk_rtc_set_AIS on
22:22:38: 3/0:22(23): trunk_rtc_gen_pattern : SIG pattern 0x0
22:22:38: 3/0:22(23): TRUNK_SC, TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
22:22:39: 3/0:13(14): lost Keepalive
22:22:39: 3/0:13(14): TRUNK_SC state : TRUNK_SC_CONN_WO_CLASS, event TRUNK_RTC_LOST_KEEPALIVE
22:22:39: 3/0:13(14): trunk_rtc_set_AIS on
22:22:39: 3/0:13(14): trunk_rtc_gen_pattern : SIG pattern 0x0
22:22:39: 3/0:13(14): TRUNK_SC, TRUNK_SC_CONN_WO_CLASS ==> TRUNK_SC_CONN_DEFAULT_IDLE
```

Note: 此输出显示发生在语音中继的两边的信令哪些使用E&M闪烁信令。使用这些同样调试信令的其他类型能被看到。如果看到正确地建立的呼叫(如显示这里)，双向音频一定存在。这可以被验证是否查看**show voice dsp**或**show call active voice brief**命令输出。如果一切查找是细致的那里，并且获得音频问题(没有音频或单程)与模拟连接，再请检查这些连接。

确定什么呼叫是UP

因为它做的很少或无益查看**show call active voice**或**show voice call summary**命令输出为中继的呼叫，您需要简单方法确定哪些语音Trunk支持激活的呼叫。其中一个最简单的方法执行此是发出**show voice trunk-conditioning signaling**命令与包括参数一道和使用ABCD作为包括的字符串，如这里所显示：

```
Phoenix#show voice trunk-conditioning signaling | include ABCD
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=1111, last-RX-ABCD=0000
!--- Timeslot 8. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=1111, last-RX-ABCD=1111 !---
Timeslot 10. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-
ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000
```

Note: 在时隙显示在时隙10和另一次呼叫的一活动呼叫开始的此输出八。如果使用它很多，您要做此相当长的命令的一个别名。

DTMF排除故障

除摘机和挂机信令外，唯一路由器通过在PBX之间的其他事(除语音以外)是DTMF音。也有音频路径，因此这通常不是问题，但是有问题。问题发生在您如何执行在该路径的音频。使用低比特率编码为了保存带宽是更有可取的。问题出来这些低比特率编码通过为人的语音被写的算法设计。除非用户使用g711编码，DTMF音不很好依照这些算法并且需要某个其他方法转达。答案在dtmf-

relay命令在。此功能允许DSP在末端，开始语音，认可DTMF音和从正常音象流分离它。基于如何配置它，DSP然后编码此语音作为不同种实时协议(RTP)信息包或作为在链路间将分开被发送的h245消息与音象流。这是在**传真中继和调制解调器中继命令**后的同一个进程。

此功能提出Trunk故障排除的另一个调试问题。如何验证什么位通过，如果没有呼叫建立，并且必须从在路由器之间的信息包流提取该信息？如何执行此取决于什么类型**dtmf-relay命令**被使用。

如此示例所显示，[dtmf-relay cisco-rtp命令](#)，使用一种所有权Cisco有效载荷类型，因此您必须查看下来DSP发现此。您能与**debug vpm port x/x**一道发出**debug vpm signal命令**：**y.z命令**(对正在考虑中的端口限制输出)发现位通过对DSP在始发端。此输出显示在始发端，不在终止端。

```
Phoenix#show voice trunk-conditioning signaling | include ABCD
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=1111, last-RX-ABCD=0000
!--- Timeslot 8. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=1111, last-RX-ABCD=1111 !---
Timeslot 10. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-
ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000
```

您能验证什么位从始发端被发送用[dtmf-relay h245-alphanumeric命令](#)。**dtmf-relay h245-alphanumeric命令**使用h.245的字母数字部分表达语音。如此示例所显示，当**debug h245 asn1命令**是启用的时，位能容易地被看到在产生和Trunk的终止端：

始发端：

```
Phoenix#show voice trunk-conditioning signaling | include ABCD
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=1111, last-RX-ABCD=0000
!--- Timeslot 8. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=1111, last-RX-ABCD=1111 !---
Timeslot 10. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-
ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-
ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000
```

终止端：

```
Phoenix#show voice trunk-conditioning signaling | include ABCD
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=0000, last-RX-ABCD=0000
last-TX-ABCD=1111, last-RX-ABCD=0000
!--- Timeslot 8. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=1111, last-RX-ABCD=1111 !---
Timeslot 10. last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-
```

ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000 last-TX-ABCD=0000, last-RX-ABCD=0000

当调试象dtmf-relay h245-alphanumeric命令使用时，[dtmf-relay h245-signal命令](#)是非常类似的，并且能被看到同样。总之，用dtmf-relay命令排除连接trunk故障是相当困难的没有被提及的调试。

[Related Information](#)

- [配置和排除透明CCS故障](#)
- [语音技术支持](#)
- [语音和 IP 通信产品支持](#)
- [Cisco IP 电话故障排除](#)
- [Technical Support - Cisco Systems](#)