

了解在IOS语音数字(T1/E1)接口上的直接拨入 (DID)

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[POTS 拨号对等体的 DID 配置](#)

[为 DID 匹配正确的入站 POTS 拨号对等体](#)

[案例研究](#)

[配置](#)

[常见问题](#)

[show 和 debug 输出示例](#)

[相关信息](#)

简介

此技术说明适用于启用了 Cisco IOS 语音的具有数字接口 (T1/E1) 的路由器/网关。有关 Cisco 模拟直接拨入 (DID) 的详细信息，请参阅：[适用于 Cisco 2600 和 Cisco 3600 系列路由器的模拟 DID](#)

注意：在大多数平台上，默认情况下在 CAS (即时、瞬间、延迟) 接口上启用 DID。因此，请不要对来电配置 direct-inward-dial 命令。在 Cisco AS5300 平台上，配置为 E & M 即时信令的接口不支持 DID。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档不限于特定的软件和硬件版本。

规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

背景信息

DID 是电话公司提供的一项服务，它使呼叫方能够直接拨打到专用交换分机 (PBX) 或语音信息包系统的分机，而无需话务员或自动呼叫助理的协助。此项服务利用 DID 中继，DID 中继仅将电话号码的最后三到五位数转发到 PBX 或路由器/网关。例如，如果公司的电话分机为 555-1000 到 555-1999 且呼叫方拨打 555-1234，则本地中心局 (CO) 会将 234 转发到 PBX 或语音信息包系统。然后，PBX 或语音信息包系统 (Cisco CallManager 和 IOS 路由器/网关) 将拨打分机 234。此整个过程对呼叫方是透明的。

在本文档中，我们将讨论两种类型的拨号对等体，如下所示：

- 普通老式电话服务 (POTS) — 这是一种通过公用交换电话网 (PSTN) 接入的传统语音呼叫，通过此网，在呼叫期间您将获取专用的 64K 电路端到端呼叫线路。POTS 拨号对等体将始终指向路由器上的语音端口
- 语音网络 — 通过数据网络的语音呼叫由多条呼叫线路组成。每条呼叫线路均在数据设备 (路由器/网关) 之间或在数据和电话设备之间 (例如路由器到 PBX) 传输。语音网络拨号对等体指向不同的目标，具体取决于所使用的网络技术。语音网络拨号对等体包括：IP 语音 (VoIP) 帧中继语音 (VoFR) ATM 语音 (VoATM) IP 多媒体邮件 (MMoIP)

当语音呼叫进入 Cisco IOS 路由器/网关时，路由器上的语音端口会被 PBX 或 CO 交换机在入站情况下捕捉。然后，路由器/网关会向呼叫方提供拨号音，并收集数字，直至可识别出站拨号对等体。不管是由人按不定期的时间间隔拨打数字，还是由发送预收集的数字的电话设备定期拨打数字，拨号对等体匹配都是按数字逐个完成的。这意味着在收到每个数字后路由器/网关均会尝试与拨号对等体匹配。此过程称为二次拨号。

不过，如果 PBX 或 CO 交换机发送包含完全路由呼叫所需的“所有”数字的设置消息，则这些数字可直接映射到出站语音网络拨号对等体。通过 DID，路由器/网关不对呼叫方提供拨号音，并且不收集数字。它会将呼叫直接转发到已配置的目标。这称为一次拨号。

上几个段落中我们讨论的路由呼叫所需的数字属于以下两种类型：

- 数字号码识别服务 (DNIS) 是一项电话公司提供的传送被叫号码 (拨打的号码) 的数字服务。
- 自动号码识别 (ANI) 是一项电话公司提供的传送主叫号码 (呼叫发起人的号码) 的数字服务。ANI 也称为呼叫线路识别 (CLID)。

POTS 拨号对等体的 DID 配置

从普通老式电话服务 (POTS) 接口接收入站呼叫时，拨号对等体的 DID 功能使路由器/网关能够使用被叫号码 (DNIS) 直接与出站拨号对等体匹配。在入站 POTS 拨号对等体上配置 DID 时，被叫号码会自动用于与出站呼叫线路的目标模式匹配。

要为 POTS 拨号对等体配置 DID，请从全局配置模式开始输入以下 Cisco IOS 命令：

```
Router(config)#dial-peer voice number pots
Router(config-dial-peer)#direct-inward-dial
```

为 DID 匹配正确的入站 POTS 拨号对等体

为使 DID 正常工作，请确保来电与在其中配置 `direct-inward-dial` 命令的正确 POTS 拨号对等体匹

配。要匹配正确的入站拨号对等体，我们建议在 DID POTS 拨号对等体下使用拨号对等体命令 `incoming called-number dnis_string`。

用于匹配拨号对等体的其他命令包括：`answer-address ani_string`、`destination-pattern string` 或 `port voice-port`。使用 `incoming called-number` 命令的优点在于每个呼叫都有关联的 DNIS 信息（被叫号码）且该命令的优先级高于前面的命令。

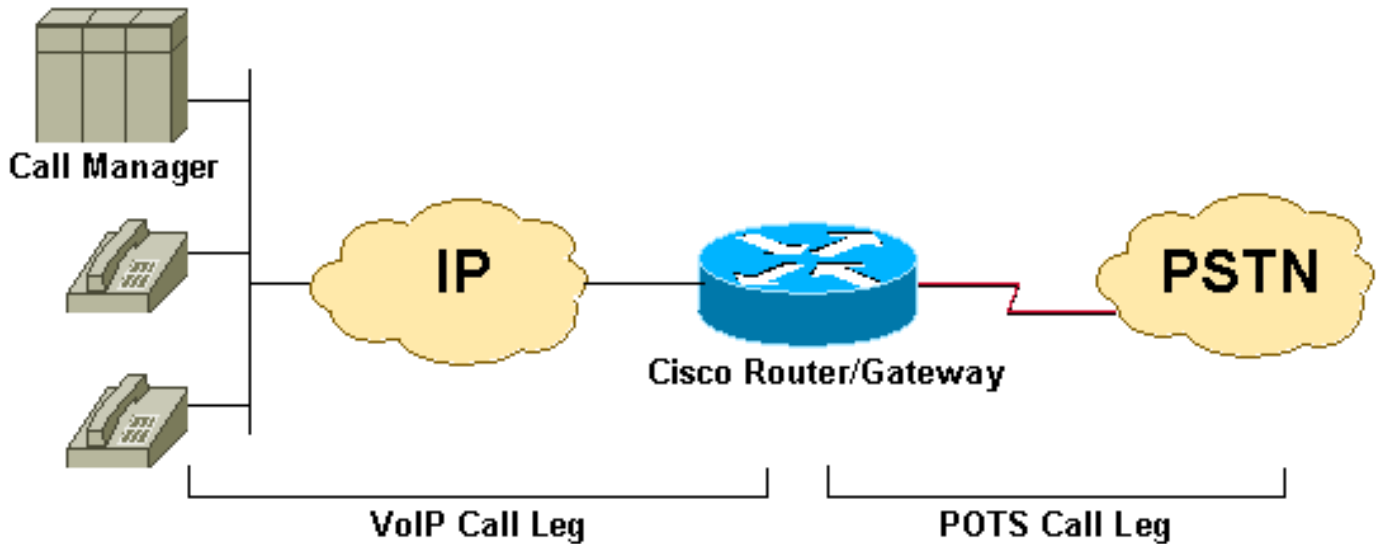
如果您不使用 `incoming called-number` 命令匹配入站拨号对等体，请考虑以下几点：

- 如果使用 ANI 信息匹配 DID POTS 拨号对等体，请确保正确配置命令 `answer-address` 且电话公司交换机提供 ANI 信息。某些 ISDN 提供商和大多数 T1 随路信令 (CAS) (功能组 D (fgd) 除外) 不提供 ANI 信息。
- 如果 `answer-address` 与 ANI 不匹配，则 ANI 可能与在其他 POTS 拨号对等体下（为出站拨号）配置的目标模式匹配。如果 `destination-pattern` 与 ANI 匹配，请确保在对应拨号对等体下配置命令 `direct-inward-dial`。
- 如果基于 `incoming called-number`、`answer-address`、`destination-pattern` 或 `port` 呼入的 DID 电话与入站 POTS 拨号对等体不匹配，则将使用默认拨号对等体 0。默认情况下在拨号对等体 0 上禁用 DID。

案例研究

使用以下示例说明上述几点。ACME 公司具有带 40 条 DID 中继（范围从 555-3100 到 555-3139）的 T1 PRI 线路。目标是将前 20 条线路分配给 Cisco IP 电话。最后 20 条线路可用于测试和将来的扩展，对于现在来说，路由器仅提供拨号音。假设 CO 交换机只发送 ISDN 设置消息中的最后五位数字，则我们可以在下表中总结上述信息。

PSTN 用户拨号	交换机发送到语音路由器/网关的数字	请使用	#中继
555-3100 到 555-3119	53100 - 53119	IP 电话的 DID 线路	20
555-3120 到 555-3139	53120 - 53139	测试和将来扩展	20



配置

注意：本示例中的某些输出省略。

```
dial-peer voice 2 pots
  destination-pattern 9T
  port 1/0:23
!--- This dial-peer is used mainly for outbound dialing with the !--- destination-pattern 9T
mapped to port 1/0:23. Note that 9 is an !--- explicit match and will be stripped. Say a call
comes from the CallManager !--- with a DNIS 914085551126, the router will send only 14085551126.
If you add !--- the dial-peer command prefix 9 or the command forward-digit all then !--- the
string 914085551126 is sent. Notice that dial-peer voice 2 pots is also !--- matched to give
dial tone to incoming users dialing this range: !--- (53120 - 53139).

dial-peer voice 3 pots
!--This dial-peer can be matched inbound only incoming called-number 5310. !--DNIS range 53100-
53109 direct-inward-dial !--If this dial-peer is matched inbound, the router is put in DID mode
! dial-peer voice 4 pots !--This dial-peer can be matched inbound only incoming called-number
5311. !--This takes care of the range 53110-53119 direct-inward-dial !--If this dial-peer is
matched inbound router is put in DID mode ! dial-peer voice 5 voip !--For our case, this dial-
peer is matched outbound only destination-pattern 53... !--When calls terminate on this router,
dial-peer 5 can be matched inbound, too. session target ipv4:172.22.1.1 !--IP address of
CallManager codec g711ulaw
```

常见问题

注意：与 debug voip ccapi inout 命令相比，断开原因代码在 debug isdn q931 命令的输出中具有不同的格式。

- 要解释来自 debug voip ccapi inout 的 Q.931 呼叫断开原因代码，请参阅：[VoIP 呼叫故障排除和调试 - 基础知识](#)
- 要解释来自 debug isdn q931 的 Q.931 呼叫断开原因代码，请参阅：[了解 debug isdn q931 断开原因代码](#)

要查看采用十进制格式的 Q.931 事件原因代码，请参阅：[ISDN 事件原因代码](#)

以下是一些症状以及可能导致这些症状的问题的示例：

- 症状：路由器/网关提供拨号音并且一直等到数字间计时器超时。然后，路由器/网关与 **debug voip ccapi inout** 原因代码 = 0x1C (号码格式无效) 或 **debug isdn q931** (对于 ISDN 接口) 断开原因代码 = 0x809C (号码格式无效) 断开连接。问题：在电话公司交换机上而不是在 Cisco IOS 路由器/网关上配置 DID。
- 症状：路由器/网关与 **debug voip ccapi inout** 原因代码 = 0x1 (号码未分配/未指定) 或 **debug isdn q931** (对于 ISDN 接口) 断开原因代码 = 0x8081 (号码未分配/未指定) 断开连接。问题：在 Cisco IOS 路由器/网关上配置了 DID 且与正确的入站 POTS 拨号对等体匹配，但设置消息不包括被叫号码 (DNIS)。在这种情况下，向电话公司确认是否为 DID 提供中继。
- 症状：路由器/网关与 **debug voip ccapi inout** 原因代码 = 0x1 (号码未分配/未指定) 或 **debug isdn q931** (对于 ISDN 接口) 断开原因代码 = 0x8081 (号码未分配/未指定) 断开连接。问题：在 Cisco IOS 路由器/网关上配置了 DID 并与其匹配，但路由器/网关上没有出站拨号对等体匹配。问题：请确保来电与在其中配置 **direct-inward-dial** 命令的正确 POTS 拨号对等体匹配。有关详细信息，请参阅本文档的“为 DID 匹配正确的入站 POTS 拨号对等体”部分。

show 和 debug 输出示例

注意：出于打印目的，以下某些调试输出行分为多个行。

```
2600#debug isdn q931
ISDN Q931 packets debugging is on
2600#debug voip ccapi inout
voip ccAPI function enter/exit debugging is on

2600#show debug
ISDN:
  ISDN Q931 packets debugging is on
  ISDN Q931 packets debug DSLs. (On/Off/No DSL:1/0/-)
  DSL  0 --> 31
  1 - - - - -
voip:
  voip ccAPI function enter/exit debugging is on

!--- Action: Cisco IOS router/gateway receives a call from the PSTN to !--- extension "53103"
*Mar 1 04:51:11.856: ISDN Se1/0:23: RX <- SETUP pd = 8 callref = 0x0001 *Mar 1 04:51:11.860:
Bearer Capability i = 0x9090A2 *Mar 1 04:51:11.860: Channel ID i = 0xA98381 *Mar 1 04:51:11.864:
Calling Party Number i = 0x0083, '408', Plan:Unknown,
  Type:Unknown
*Mar 1 04:51:11.868:          Called Party Number i = 0x80, '53103', Plan:Unknown,
  Type:Unknown
!--- ISDN Q.931 and Voip ccapi inout debugs collectively show a DNIS of 53103 and !--- an ANI
(Automatic Number Identification) of 408 sent in unknown plan and type. *Mar 1 04:51:11.880:
cc_api_call_setup_ind (vdbPtr=0x831721D8, callInfo=
{called=53103,called_oct3=0x80,calling=408,calling_oct3=0x0, calling_oct3a=0x83,
calling_xlated=false,subscriber_type_str=RegularLine, fdest=1,peer_tag=3,
prog_ind=0},callID=0x83349DF8)
*Mar 1 04:51:11.884: cc_API_call_setup_ind type 13 , prot 0
*Mar 1 04:51:11.888: cc_process_call_setup_ind (event=0x83149130)
*Mar 1 04:51:11.888: >>>>CCAPI handed cid 41 with tag 3 to app "DEFAULT"
!--- POTS dial-peer 3 was matched inbound *Mar 1 04:51:11.888: sess_appl:
ev(24=CC_EV_CALL_SETUP_IND), cid(41), disp(0) *Mar 1 04:51:11.888: sess_appl:
ev(SSA_EV_CALL_SETUP_IND), cid(41), disp(0) *Mar 1 04:51:11.888: ssaCallSetupInd *Mar 1
04:51:11.892: ccCallSetContext (callID=0x29, context=0x83303C00)
!--- The POTS leg is created and assigned a callid of 0x29 *Mar 1 04:51:11.892: ssaCallSetupInd
cid(41), st(SSA_CS_MAPPING),oldst(0), ev(24)ev->e.evCallSetupInd.nCallInfo.finalDestFlag = 1
*Mar 1 04:51:11.892: ssaCallSetupInd finalDest cllng(408), cllcd(53103)
!--- Due to the direct-inward-dial config under dial-peer 3, the DNIS sent in !--- the setup
```

```

request is considered sufficient to match an outbound dial-peer. !--- This is clear with flag
set to 1. *Mar 1 04:51:11.892: ssaCallSetupInd cid(41), st(SSA_CS_CALL_SETTING),oldst(0),
ev(24)dpMatchPeersMoreArg result= 0 *Mar 1 04:51:11.892: ssaSetupPeer cid(41) peer list: tag(5)
called number (53103) !--- Dial-peer table lists only dial-peer 5 as matched outbound against
the DNIS. *Mar 1 04:51:11.892: ssaSetupPeer cid(41), destPat(53103), matched(2), prefix(),
peer(83369DB8), peer->encapType (2) !--- Due to destination-pattern having 2 digits and 3 dots,
explicit match is !--- reported as 2. *Mar 1 04:51:11.896: ccCallProceeding (callID=0x29,
prog_ind=0x0) *Mar 1 04:51:11.896: ccCallSetupRequest (Inbound call = 0x29, outbound peer =5,
dest=, params=0x831578C0 mode=0, *callID=0x83157C28, prog_ind = 0) *Mar 1 04:51:11.896:
ccCallSetupRequest numbering_type 0x80 *Mar 1 04:51:11.896: dest pattern 53..., called 53103,
digit_strip 0
*Mar 1 04:51:11.896: callingNumber=408, calledNumber=53103, redirectNumber=
display_info= calling_oct3a=83
!--- Just before matching an outbound dial-peer, we remember that we have !--- seen the same ANI
and DNIS in the ISDN setup and in the ccapi debug initially. !--- In other words, the router did
not collect additional digits after the seizure. !--- Equal value of DNIS at setup request and
before matching an outbound !--- dial-peer is the whole purpose of DID *Mar 1 04:51:11.896:
accountNumber=, finalDestFlag=1, guid=c66d.980c.17a8.0051.0000.0000.010a.998a *Mar 1
04:51:11.896: peer_tag=5 *Mar 1 04:51:11.896: ccIFCallSetupRequestPrivate: (vdbPtr=0x824C6344,
dest=, callParams={called=53103,called_oct3=0x80, calling=408,calling_oct3=0x0,
calling_xlated=false,subscriber_type_str=RegularLine, fdest=1, voice_peer_tag=5},mode=0x0)
vdbPtr type = 3 *Mar 1 04:51:11.900: ccIFCallSetupRequestPrivate: (vdbPtr=0x824C6344, dest=,
callParams={called=53103, called_oct3 0x80, calling=408,calling_oct3 0x0, calling_xlated=false,
fdest=1, voice_peer_tag=5}, mode=0x0, xltrc=-5) *Mar 1 04:51:11.900: ccSaveDialpeerTag
(callID=0x29, dialpeer_tag= *Mar 1 04:51:11.900: ccCallSetContext (callID=0x2A,
context=0x8330408C) *Mar 1 04:51:11.900: ccCallReportDigits (callID=0x29, enable=0x0) *Mar 1
04:51:11.904: cc_API_call_report_digits_done (vdbPtr=0x831721D8, callID=0x29, disp=0) *Mar 1
04:51:11.904: sess_appl: ev(52=CC_EV_CALL_REPORT_DIGITS_DONE), cid(41), disp(0) *Mar 1
04:51:11.904: cid(41)st(SSA_CS_CALL_SETTING)ev (SSA_EV_CALL_REPORT_DIGITS_DONE)
oldst(SSA_CS_MAPPING)cfid(-1)csize(0)in(1)fDest(1) . !--- Output Omitted . !--- The following
output displays the Call is finished *Mar 1 04:51:52.442: ISDN Se1/0:23: RX <- DISCONNECT pd = 8
callref = 0x0001 *Mar 1 04:51:52.442: Cause i = 0x8290 - Normal call clearing *Mar 1
04:51:52.458: ISDN Se1/0:23: TX -> RELEASE pd = 8 callref = 0x8001 *Mar 1 04:51:52.458:
cc_API_call_disconnected(vdbPtr=0x831721D8, callID=0x29,
cause=0x10)
*Mar 1 04:51:52.462: sess_appl: ev(11=CC_EV_CALL_DISCONNECTED), cid(41), disp(0)
*Mar 1 04:51:52.462: cid(41)st(SSA_CS_ACTIVE)ev(SSA_EV_CALL_DISCONNECTED)
oldst(SSA_CS_ACTIVE)cfid(9)csize(2)in(1)fDest(1)
*Mar 1 04:51:52.462: -cid2(42)st2(SSA_CS_ACTIVE)oldst2(SSA_CS_ALERT_RCVD)
*Mar 1 04:51:52.462: ssa: Disconnected cid(41) state(5) cause(0x10)
*Mar 1 04:51:52.462: ccConferenceDestroy (confID=0x9, tag=0x0)
*Mar 1 04:51:52.462: cc_API_bridge_drop_done (confID=0x9, srcIF=0x824C6344,
srcCallID=0x2A, dstCallID=0x29, disposition=0 tag=0x0)
*Mar 1 04:51:52.466: cc_API_bridge_drop_done (confID=0x9, srcIF=0x831721D8,
srcCallID=0x29, dstCallID=0x2A, disposition=0 tag=0x0)
*Mar 1 04:51:52.466: sess_appl: ev(30=CC_EV_CONF_DESTROY_DONE), cid(41), disp(0)
*Mar 1 04:51:52.470: cid(41)st(SSA_CS_CONF_DESTROYING)ev(SSA_EV_CONF_DESTROY_DONE)
oldst(SSA_CS_ACTIVE)cfid(-1)csize(2)in(1)fDest(1)
*Mar 1 04:51:52.470: -cid2(42)st2(SSA_CS_CONF_DESTROYING)oldst2(SSA_CS_ALERT_RCVD)
*Mar 1 04:51:52.470: ssaConfDestroyDone
*Mar 1 04:51:52.470: ccCallDisconnect (callID=0x29, cause=0x10 tag=0x0)
*Mar 1 04:51:52.470: ccCallDisconnect (callID=0x2A, cause=0x10 tag=0x0)

!--- These two lines are great for finding the source of the disconnect. !--- They tell us that
the first call leg with callid 0x29 (POTS call leg) !--- disconnected with cause code 0x10. So
either the end POTS user hung up or the !--- telephony equipment disconnected unintentionally.
From the router's point of !--- view, both are the same. *Mar 1 04:51:52.470: ISDN Se1/0:23: RX
<- RELEASE_COMP pd = 8 callref = 0x0001 *Mar 1 04:51:52.499:
cc_API_call_disconnect_done(vdbPtr=0x831721D8, callID=0x29, disp=0, tag=0x0) !--- Debug
truncated here 2600#show call active voice brief
!--- This show command is good to verify which are the dial-peers matched by the !--- call. In
the example below, the output show the POTS call-leg matched !--- dial-peer voice 3 pots (pid:3)
the VoIP call-leg matched !--- dial-peer voice 5 voip (pid:5). !--- some output omitted

```

Total call-legs: 2

3A : 799622hs.1 +112 **pid:3** Answer 408 active

dur 00:00:07 tx:385/61600 rx:160/23690

Tele 1/0:23:33: TX:7730/3060/0ms g711ulaw noise:-42 acom:0 i/0:-43/-53 dBm

3A : 799625hs.1 +106 **pid:5** Originate 53103 active

dur 00:00:07 TX:160/23690 rx:385/61600

IP 171.68.168.250:25704 rtt:0ms pl:4980/0ms lost:0/0/0 delay:64/64/65ms g711ulaw

[相关信息](#)

- [了解 Cisco IOS 平台上的拨号对等体和呼叫线路](#)
- [了解 Cisco IOS 平台上的入站和出站拨号对等体](#)
- [ISDN 交换机类型、代码和值](#)
- [配置拨号计划、拨号对等体和数字操作](#)
- [对 Cisco 3600 系列配置 IP 语音](#)
- [了解 2 端口直接拨入 \(2 DID \) 语音接口卡](#)
- [技术支持 - Cisco Systems](#)