

# 瞭解IOS語音數位(T1/E1)介面上的直接撥入(DID)

## 目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[慣例](#)

[背景資訊](#)

[POTS撥號對等體的DID配置](#)

[為DID匹配正確的入站POTS撥號對等體](#)

[案例研究](#)

[組態](#)

[常見問題](#)

[顯示和調試輸出示例](#)

[相關資訊](#)

## 簡介

本技術說明適用於具有數字介面(T1/E1)的Cisco IOS語音啟用路由器/網關。有關思科模擬直接撥入(DID)的詳細資訊，請參閱：[Cisco 2600和Cisco 3600系列路由器的模擬DID](#)

**注意：**在大多數平台上，CAS（即時、閃爍、延遲）介面預設啟用DID。因此，請勿為傳入呼叫配置direct-inward-dial命令。在Cisco AS5300平台上，為E & M即時信令配置的介面不支援DID。

## 必要條件

### 需求

本文件沒有特定需求。

### 採用元件

本文件所述內容不限於特定軟體和硬體版本。

### 慣例

如需文件慣例的詳細資訊，請參閱[思科技術提示慣例](#)。

## 背景資訊

DID是一項由電話公司提供的服務，呼叫者無需操作員或自動呼叫助理的協助，即可直接撥打專用交換機(PBX)或資料包語音系統上的分機。此服務使用DID中繼，只將電話號碼的最後三到五位數字轉發到PBX或路由器/網關。例如，如果某公司的電話分機號為555-1000到555-1999，而呼叫者撥打555-1234，則本地中央辦公室(CO)會將234轉發到PBX或資料包語音系統。然後PBX或資料包語音系統 ( Cisco CallManager和IOS路由器/網關 ) 將振鈴呼叫分機234。整個過程對呼叫方是透明的。

在本文檔中，我們將討論以下兩種型別的撥號對等體：

- 普通舊式電話服務(POTS) — 這是通過公共交換電話網路(PSTN)進行的傳統語音呼叫，在此呼叫期間，您將獲得專用的64K電路端到端呼叫支路。POTS撥號對等體將始終指向路由器上的語音埠
- 語音網路 — 通過資料網路的語音呼叫由多個呼叫段組成。每個呼叫支路可在資料裝置 ( 路由器/網關 ) 之間或在資料和電話裝置 ( 例如路由器到PBX ) 之間傳輸。語音網路撥號對等體根據所使用的網路技術指向不同的目的地。語音網路撥號對等體包括：IP語音(VoIP)透過框架轉送傳輸的語音(VoFR)透過ATM傳輸的語音(VoATM)透過IP傳輸的多媒體郵件(MoIP)

當語音呼叫進入Cisco IOS路由器/網關時，PBX或CO交換機將獲取路由器上的語音埠。然後，路由器/網關向呼叫方顯示撥號音，並收集數字，直到能夠識別出站撥號對等體。無論數字是由人以不規則的間隔撥號，還是由傳送預先收集的數字的電話裝置以規則的方式撥號，撥號對等體匹配都是逐位完成的。這表示路由器/網關在收到每個數字後嘗試匹配撥號對等體。此過程稱為兩階段撥號。

但是，如果PBX或CO交換機傳送設定消息，其中包含完全路由呼叫所必需的所有數字，則這些數字可以直接對映到出站語音網路撥號對等體。使用DID時，路由器/網關不會向呼叫方顯示撥號音，也不會收集數字。它將呼叫直接轉接到已配置的目的地。這稱為一級撥號。

路由上述段落中討論的呼叫所必需的數字位為以下兩種型別：

- 數字號碼識別服務(DNIS)是電信公司提供的數位服務，用於傳送被叫號碼 ( 被撥打的號碼 )。
- 自動號碼識別(ANI)是一種電信公司提供的數位服務，用於傳送主叫號碼 ( 呼叫發起者的號碼 )。ANI也稱為呼叫線路識別(CLID)。

## POTS撥號對等體的DID配置

從普通舊式電話服務(POTS)介面接收入站呼叫時，撥號對等體中的DID功能使路由器/網關能夠使用被叫號碼(DNIS)直接匹配出站撥號對等體。當在入站POTS撥號對等體上配置DID時，被叫號碼將自動用於匹配出站呼叫段的目標模式。

要為DID配置POTS撥號對等體，請輸入以下Cisco IOS命令，從全域性配置模式開始：

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

## 為DID匹配正確的入站POTS撥號對等體

為使DID正常工作，請確保傳入呼叫與配置了命令direct-inward-dial的正確POTS撥號對等體匹配。要匹配正確的入站撥號對等體，建議在DID POTS撥號對等體下使用dial peer命令incoming called-number dnis\_string。

用於匹配撥號對等體的其他命令包括：answer-address ani\_string、destination-pattern string 或

port *voice-port*。使用 *incoming called-number* 命令的優點是，每個呼叫都有關聯的DNIS資訊 (called-number)，並且優先於先前的命令。

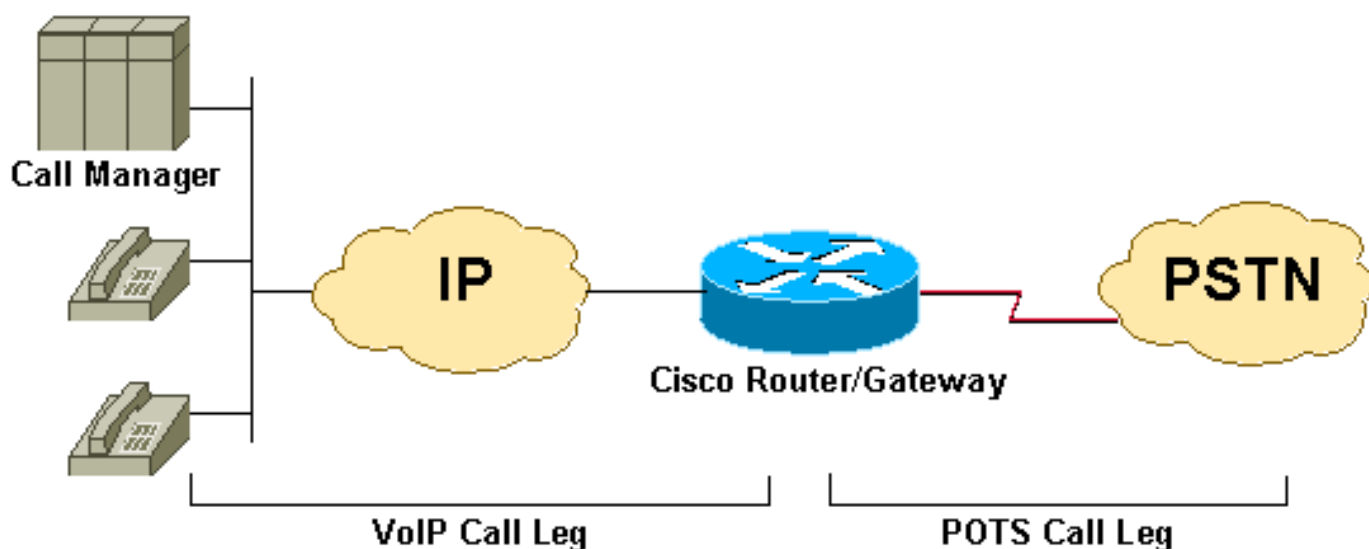
如果不使用 *incoming called-number* 命令匹配入站撥號對等體，請考慮以下事項：

- 如果使用ANI資訊匹配DID POTS撥號對等體，請確保正確配置命令 *answer-address*，並且電信交換機提供ANI資訊。有些ISDN提供者和大多數T1通道關聯訊號(CAS)(功能群組D(fgd)除外)不提供ANI資訊。
- 如果 *answer-address* 與ANI不匹配，則ANI可能匹配另一個POTS撥號對等體下配置的 *destination-pattern* (用於出站撥號)。如果 *destination-pattern* 與ANI匹配，請確保在該撥號對等體下配置命令 *direct-inward-dial*。
- 如果根據 *incoming called-number* 或 *answer-address* 或 *destination-pattern* 或 *port* 將傳入DID呼叫與傳入POTS撥號對等體不匹配，則將使用預設撥號對等體0。撥號對等體0上預設禁用DID。

## 案例研究

使用以下示例說明上述幾點。ACME公司擁有帶有40個DID中繼的T1 PRI線路，範圍從555-3100到555-3139。目標是將前20條線路分配給思科IP電話。最後20條線路可供測試、將來擴展使用，目前路由器只提供撥號音。假定CO交換機只傳送ISDN設定消息的最後五位數字，我們可以在下表中總結上述資訊。

PSTN使用者撥號	交換機傳送到語音路由器/網關的數字	使用	中繼數量
555-3100至555-3119	53100 - 53119	IP電話的DID線路	20
555-3120至555-3139	53120 - 53139	測試和未來擴展	20



## 組態

注意：本示例中的一些輸出被省略。

```
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
```

## 常見問題

註：與debug voip ccapi inout命令相反，debug isdn q931輸出中的斷開連線原因代碼具有不同的格式。

- 要解釋debug voip ccapi inout中的Q.931呼叫斷開原因代碼，請參閱：[VoIP呼叫故障排除和調試 — 基本知識](#)
- 要解釋debug isdn q931的Q.931呼叫斷開原因代碼，請參閱：[瞭解debug isdn q931結束通話原因代碼](#)

要檢視Q.931事件原因代碼十進位制格式，請參閱：[ISDN事件原因代碼](#)

以下是一些症狀示例以及可能導致這些症狀的問題：

- 症狀：路由器/網關提供撥號音並等待數字間計時器超時。然後斷開連線，並使用debug voip ccapi inout原因代碼= 0x1C (無效數字格式) 或debug isdn q931 (用於ISDN介面) 斷開原因代碼= 0x809C (無效數字格式)。問題：DID是在電信交換機上配置的，而不是在Cisco IOS路由器/網關上配置的。
- 症狀：路由器/網關斷開與debug voip ccapi inout原因代碼= 0x1 (未分配/未分配編號) 或debug isdn q931 (對於ISDN介面) 斷開原因代碼= 0x8081 (未分配/未分配編號) 的連線。問題：已配置DID，且在Cisco IOS路由器/網關上匹配正確的入站POTS撥號對等體，但設定消息不包括被叫號碼(DNIS)。在這種情況下，向電信公司驗證是否已為DID調配中繼。
- 症狀：路由器/網關斷開與debug voip ccapi inout 原因代碼= 0x1 (未分配/未分配編號) 或debug isdn q931 (對於ISDN介面) 斷開原因代碼= 0x8081 (未分配/未分配編號) 的連線。問題：DID在Cisco IOS路由器/網關上配置並匹配，但是在路由器/網關上沒有出站撥號對等體匹配。問題：確保傳入呼叫與配置了命令direct-inward-dial的正確POTS撥號對等體匹配。有關詳細資訊，請參閱本文檔的為DID匹配正確的入站POTS撥號對等體部分

## 顯示和調試輸出示例

註：以下調試輸出行的某些行被分成多行進行列印。

```
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)csz(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)csz(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)csz(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

```

## 相關資訊

- [瞭解Cisco IOS平台上的撥號對等體和呼叫段](#)
- [瞭解Cisco IOS平台上的傳入和傳出撥號對等體](#)
- [ISDN交換器型別、代碼和值](#)
- [配置撥號計畫、撥號對等體和數字操控](#)
- [為Cisco 3600系列配置IP語音](#)
- [瞭解2埠直接撥入\(2 DID\)語音介面卡](#)
- [技術支援 - Cisco Systems](#)