Understand Direct-Inward-Dial (DID) on IOS Voice Digital (T1/E1) Interfaces

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Background Information

DID Configuration for POTS Dial Peers

Matching the Correct Inbound POTS Dial Peer for DID

Case Study

Configuration

Common Issues

Sample Show and Debug Output

Introduction

This document describes Cisco IOS® voice-enabled routers/gateways with digital interfaces (T1/E1). For more information on Cisco analog Direct Inward Dialing (DID) refer to: Analog DID for Cisco 2600 and Cisco 3600 Series Routers

Note: On most platforms, DID is enabled by default on CAS (immediate, wink, delay) interfaces. Therefore, do not configure the direct-inward-dial command for incoming calls. On Cisco AS5300 platforms, DID is not supported on interfaces configured for E & M immediate signaling.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

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, ensure that you understand the potential impact of any command. This document is not restricted to specific software and hardware versions.

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Background Information

DID is a service offered by telephone companies that enables callers to dial directly to an extension on a Private Branch Exchange (PBX) or packet voice system without the assistance of an operator or automated call attendant. This service makes use of DID trunks, which forward only the last three to five digits of a phone number to the PBX or router/gateway. If for example, a company has phones extensions 555-1000 to 555-1999, and a caller dials 555-1234, the local central office (CO) would forward 234 to the PBX or packet voice system. The PBX or packet voice system (Cisco CallManager and IOS router/gateway) would then ring extension 234. This entire process is transparent to the caller.

In this document, we discuss two types of dial peers as follows:

- Plain old telephone service (POTS)—This is a traditional voice call placed over the Public Switched Telephone Network (PSTN) where you get a dedicated 64K circuit end-to-end call leg during the duration of the call. POTS dial peers will always point to a voice-port on the router
- Voice-Network—A voice call over the data network is made up of several call-legs. Each call-leg either travels between data devices (routers/gateways) or between data and telephony devices (such as a router to a PBX). Voice-Network dial peers point to different destinations depending on the network technology used. Voice-Network dial peers include the following:
    - Voice over IP (VoIP)
    - Voice over Frame Relay (VoFR)
    - Voice over ATM (VoATM)
    - Multimedia Mail over IP (MMoIP)

When a voice call comes into the Cisco IOS router/gateway, the voice port on the router is seized inbound by a PBX or CO switch. The router/gateway then presents a dial tone to the caller and collects digits until it can identify an outbound dial peer. Whether the digits are dialed with irregular intervals by humans or in a regular fashion by telephony equipment sending the pre-collected digits, dial-peer matching is done digit-by-digit. This means the router/gateway attempts to match a dial peer after each digit is received. This process is called two-stage dialing.

However, if the PBX or CO switch sends a setup message containing "all" the digits necessary to fully route the call, those digits can be mapped to an outbound Voice-Network dial-peer directly. With DID, the router/gateway does not present a dial tone to the caller and does not collect digits. It forwards the call directly to the configured destination. This is called one-stage dialing.

The digits necessary to route the calls that we discussed in the above paragraphs are of the two following types:

- Digital Number Identification Service (DNIS) is a telco-provided digital service that delivers the called-number (the number that is dialed).
- Automatic Number Identification (ANI) is a telco-provided digital service that delivers the calling-number (the number of the call originator). ANI is also referred to as Calling Line Identification (CLID).

DID Configuration for POTS Dial Peers

When receiving an inbound call from a plain old telephone service (POTS) interface, the DID feature in dial

peers enables the router/gateway to use the called number (DNIS) to directly match an outbound dial peer. When DID is configured on the inbound POTS dial peer, the called number is automatically used to match the destination pattern for the outbound call leg.

To configure a POTS dial peer for DID, enter the following Cisco IOS commands beginning in global configuration mode:

```
<#root>

Router(config)#

dial-peer voice number pots

Router(config-dial-peer)#

direct-inward-dial
```

Matching the Correct Inbound POTS Dial Peer for DID

For DID to work correctly, make sure the incoming call matches the correct POTS dial-peer where the command **direct-inward-dial** is configured. To match the correct inbound dial peer, we recommend using the dial peer command **incoming called-number dnis_string** under the DID POTS dial peer.

Other commands used to match dial-peers include: **answer-address ani_string** , **destination-pattern string** or **port voice-port** . The advantage of using the **incoming called-number** command is that every call has associated DNIS information (called-number) and it has priority over the previous commands.

If you do not use the **incoming called-number** command to match the inbound dial peer, consider the following:

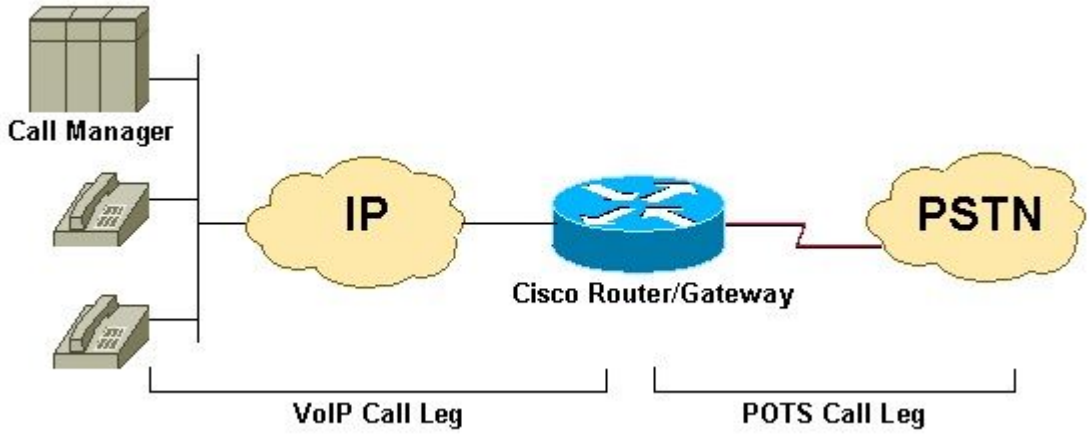
- If using ANI information to match the DID POTS dial-peer, make sure the command **answer-address** is configured correctly and the telco-switch is providing ANI information. Some ISDN providers and most T1 channel associated signaling (CAS), except Feature Group D (fgd), do not provide ANI information.
- If **answer-address** is NOT matched against ANI, then the ANI may be matching the **destination-pattern** configured (for outbound dialing) under another POTS dial-peer. If the **destination-pattern** is matched against ANI, make sure that the command **direct-inward-dial** is configured under that dial-peer.
- If the incoming DID call is not matched to an inbound POTS dial-peer based on **incoming called-number** or **answer-address** or **destination-pattern** or **port**, then default dial-peer 0 will be used. DID is disabled by default on dial-peer 0.

Case Study

Use the following example, to illustrate the above points. ACME Company has T1 PRI lines with 40 DID trunks in the range of 555-3100 to 555-3139. The goal is to assign the first 20 lines to Cisco IP phones. The last 20 lines are available for testing, future expansion and for now the router gives dial-tone only. Assuming that the CO switch is sending only the last five digits in the ISDN set up message, we can summarize the above information in the following table.

PSTN Users Dial	Digits Sent by Switch to Voice	Use	#

	Router/Gateway		Trunks
555-3100 to 555-3119	53100 - 53119	DID lines for IP Phones	20
555-3120 to 555-3139	53120 - 53139	Testing and future expansion	20



Configuration

Note: Some of the output in this example is omitted.

```
<#root>

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 mapp

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
```

Common Issues

Note: Disconnect cause codes have different formats in the output of the **debug isdn q931** as opposed to the **debug voip ccapi inout** command.

- To interpret the Q.931 call disconnect cause codes from **debug voip ccapi inout** refer to: Troubleshoot & Debug VoIP Calls - the Basics
- To interpret the Q.931 call disconnect cause codes from **debug isdn q931** refer to: Understanding debug isdn q931 Disconnect Cause Codes

To see the Q.931 event cause codes in decimal format refer to: ISDN event cause codes

The following are some examples of symptoms and the issues which may cause them:

- Symptom: Router/gateway provides dial-tone and waits until the interdigit timer times out. Then it disconnects with the **debug voip ccapi inout** cause code = 0x1C (invalid number format) or **debug isdn q931** (for ISDN interfaces) disconnect cause code = 0x809C (invalid number format).
    - Issue: DID is configured on the telco switch but not on the Cisco IOS router/gateway.
- Symptom: Router/gateway disconnects with the **debug voip ccapi inout** cause code = 0x1 (unallocated / unassigned number) or **debug isdn q931** (for ISDN interfaces) disconnect cause code = 0x8081 (unallocated / unassigned number).
    - Issue: DID is configured and the correct inbound POTS dial peer is matched on the Cisco IOS router/gateway, but the setup message does not include called-number (DNIS). In this case verify with the telco that the trunk is provisioned for DID.
- Symptom: Router/gateway disconnects with the **debug voip ccapi inout** cause code = 0x1 (unallocated/unassigned number) or **debug isdn q931** (for ISDN interfaces) disconnect cause code =

0x8081 (unallocated/unassigned number).

- Issue: DID is configured and matched on the Cisco IOS router/gateway, but there is no outbound dial-peer match on the router/gateway.
- Issue: Make sure the incoming call matches the correct POTS dial-peer where the command **direct-inward-dial** is configured. For more information refer to the Matching the Correct Inbound POTS Dial Peer for DID section of this document

Sample Show and Debug Output

Note: Some of the following debug output lines are broken into multiple lines for printing purposes.

```
<#root>

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 (Aut


*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), clled(53103)

!--- Due to the direct-inward-dial config under dial-peer 3, the DNIS sent in !--- the setup reque


*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 a


*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 th

```
*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

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
```