Caller ID Name Delivery Issues on Cisco IOS Gateways

Document ID: 23444

Contents

Introduction
Prerequisites
Requirements
Components Used
Conventions
Configure
Network Diagram
Configurations
How to Configure SIP Extensions for Caller Identity
Verify
Troubleshoot
Troubleshooting Debugs and Analyzing Traces
Related Information

Introduction

Caller ID is an analog service by which a telephone central office (CO) switch sends digital information about the incoming call. The Caller ID name delivery feature for analog Foreign Exchange Station (FXS) ports was first introduced in Cisco IOS® Software Release 12.1(2)XH and is available on all later Cisco IOS software releases. This feature is available and configurable on a per−port basis to phones connected to analog FXS voice ports. This feature is also available on analog Foreign Exchange Office (FXO).

Note: FXS ports transmit Caller ID, while FXO ports receive Caller ID. The Caller ID interoperates with analog phones, public switched telephone networks (PSTNs), private branch exchanges (PBXs), H.323 terminals (such as Microsoft NetMeeting), Cisco CallManager, and IP phones. Therefore, Caller ID can be delivered across a telephony network that consists of all or some of these devices, with some exceptions.

Furthermore, there is a Cisco IOS feature that allows a network designer to block Caller ID from being transmitted from the FXS port, if necessary. The Caller ID is unblocked, by default, for all calls; but, Caller ID can be blocked on a per−port basis. When you turn on this feature on any given port, it blocks the Caller ID of all calls that originate from that port.

Prerequisites

Requirements

Before you attempt this configuration, make sure that you understand the command references for this feature, which are described here:

- **[no] caller−id enable** Enables and disables Caller ID. Default is Caller ID disabled. This enables or disables the transmission of Caller ID on an FXS port and enables or disables reception of Caller ID on an FXO port.
- **[no] station−id number***string* Provides a station number to use as the calling number associated with the voice port. The *string* parameter is optional and, if provided, is passed as the calling number

when a call originates from this voice port. If this parameter is not specified, the calling number attained from a reverse−dial−peer search is used. If no Caller ID is received on an FXO voice port, this parameter is used as the calling number. The maximum number of characters that can be used for the *string* parameter is 15 characters.

- **[no] station−id name***string* Provides a station name associated with the voice port. The *string* parameter is passed as the calling name to the remote end when a call originates from this voice port. If no Caller ID is received on an FXO voice port, this parameter is used as the calling name. The maximum number of characters that can be used for the *string* parameter is 15 characters.
- **[no] caller−id block** Blocks or unblocks Caller ID. The default is Caller ID unblocked. This command blocks or unblocks the Caller ID of all calls that originate from this port. This command is available only on FXS voice ports.
- **[no] ring number** *string* This command sets the maximum number of rings to be detected before a call is answered over an FXO voice port. The ring number command is how Cisco receives the Caller ID information after two rings. For more information, refer the ring number section of Cisco IOS Voice Command Reference.

Components Used

This configuration was developed and tested with these software and hardware versions:

- Cisco 2600 IOS® routers with Ethernet card, analog FXS card, NM−2V module and VWIC−MFT vice−card with NM−HDV module
- A simple analog phone with RJ−11 connected to one Cisco 2600
- Any third−party vendor PBX with a T1 interface for other Cisco 2600s
- Cisco IOS versions used in the 2600s are mainline Cisco IOS® Software Release 12.2(10)

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.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Configure

In this section, you are presented with the information to configure the features described in this document.

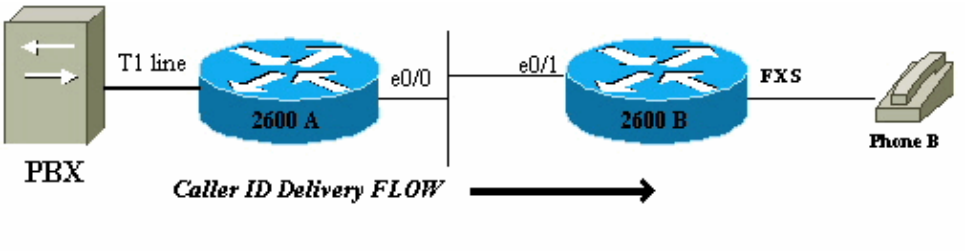
Note: In order to find additional information on the commands used in this document, use the Command Lookup Tool (registered customers only) .

Network Diagram

The simple telephony network in this diagram shows an example of Caller ID delivery through the LAN and the FXS port on the Cisco 2600 B side to Phone B. The Caller ID is *not* supported on E&M channel associated signaling (CAS) lines. In this example, Caller ID is spoofed as though it came from the CAS line, in order to send it to the FXS port. For digital lines, only ISDN lines support Caller ID delivery by default, and CAS type **fgd** is the only CAS type to support Caller ID delivery.

On the Cisco AS5300 and AS5800 platforms, a feature of CAS signaling Feature Group B (FGB) allows Automatic Number Identification (ANI) to be received upon configuration of the T1. If this signaling is used, the Caller ID is automatically received on the Cisco 5300 or 5800. This feature is explained further in CAS on T1 Voice Channels.

This configuration shows only the elements that pertain to Voice over IP (VoIP) and Caller ID commands:



The call flow is from PBX to Phone B. In this scenario, if a call comes in to 2600 A and is delivered out to 2600 B, then the Caller ID display on Phone B is:

```
Name = Outside
CallingNumber = 5553030
Time = 2600 B s local clock setting
```

Configurations

This document uses these configurations:

- Cisco 2600 A
- Cisco 2600 B

Cisco 2600 A

```
!
Controller T1 1/0
framing esf
linecode b8zs
ds0-group 1 timeslots 1-4 type e&m-wink-start
!
interface ethernet 0/0
ip address 10.10.1.2 255.255.255.0
!
voice-port 1/0:1
station-id name Outside

!--- Command line interface (CLI) to spoof
!--- Name Display on phone for all calls
!--- from CAS line.

station-id number 5553030

!--- CLI to spoof Number Display on phone
!--- for all calls from CAS line.

!
dial-peer voice 9913050 voip
destination-pattern 9913050
session target ipv4:10.10.1.1
!
```

Cisco 2600 B

```
!
interface ethernet 0/1
ip address 10.10.1.1 255.255.255.0
!
voice-port 1/1/0
```

```
caller-id enable

!--- Enables Caller ID feature.

!
dial-peer voice 100 pots
destination-pattern 9913050
port 1/1/0
!
```

How to Configure SIP Extensions for Caller Identity

In order to enable translation of the SIP header Remote−Party−ID, use the **remote−party−id** command in SIP UA configuration mode.

```
Router(config)#sip-ua

Router(config-sip-ua)#remote-party-id
```

When the **remote−party−id** command is enabled, if a Remote−Party−ID header is present in the incoming INVITE message, the calling name and number extracted from the Remote−Party−ID header are sent as the calling name and number in the outgoing setup message. For more information about SIP Extensions for Caller Identity, refer to SIP Extensions for Caller Identity and Privacy.

Verify

For verification and basic configurations of Caller ID, refer to CAS on T1 Voice Channels.

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

Troubleshooting Debugs and Analyzing Traces

You can turn on several debugs in order to troubleshoot the Caller ID feature on the routers. The voice port module (VPM) signaling debugs (**debug vpm signal**) track the standard fxs−loopstart debugs with the Caller ID feature turned on. These debugs are analyzed from the perspective of the terminating router and the FXS port of that router; the Caller ID is received on this end.

Debugs from Terminating Gateway 2600 B on the FXS Port

```
2600B# show debug
Voice Port Module signaling debugging is on
Nov 17 17:05:27.144 EST: [1/1/0, FXSLS_ONHOOK, E_HTSP_SETUP_REQ] fxsls_onhook_setup
Nov 17 17:05:27.144 EST: [1/1/0] set signal state = 0x0 timestamp= 0 htsp_progress
Nov 17 17:05:27.144 EST: [1/1/0] set signal state = 0x0 timestamp= 0

!--- Here is what is delivered to the phone.

Nov 17 17:05:27.144 EST: [1/1/0] htsp_set_caller_id_tx calling num=5553030
   display_info=Outside called num=9913050

!--- Here is the Hex that is sent out to the phone.

Nov 17 17:05:27.144 EST: [1/1/0] Caller ID String 80 1C
   01 08 31 31 31 37 32 32 30 35 07 35 35 35 33 30 33 30
   07 07 4F 75 74 73 69 64 65 6F
```

```
Nov 17 17:05:27.148 EST: [1/1/0] htsp_set_caller_id_tx Caller ID: FSK_DURING_RING
Nov 17 17:05:27.148 EST: [1/1/0] htsp_start_caller_id_tx string length=31
Nov 17 17:05:27.160 EST: [1/1/0, FXSLS_WAIT_OFFHOOK, E_HTSP_VOICE_CUT_THROUGH]
   fxsls_waitoff_voice
Nov 17 17:05:34.836 EST: [1/1/0, FXSLS_WAIT_OFFHOOK, E_HTSP_RELEASE_REQ]
   fxsls_waitoff_release
Nov 17 17:05:34.836 EST: [1/1/0] set signal state = 0x4 timestamp = 0
```

Note: Lines of this output that are on more than one line are actually displayed as one line in the debug output.

This is displayed on Phone B:

```
CallerID = 5553030
Name = Outside
Time = 10:05P Nov17

!--- Time is received from the Local Router Clock.
```

When the hexadecimal Caller ID String is decoded in the example, it provides these results:

```
Nov 17 17:05:27.144 EST: [1/1/0] Caller ID String 80 1C
   01 08 31 31 31 37 32 32 30 35
   02 07 35 35 35 33 30 33 30
   07 07 4F 75 74 73 69 64 65 6F

!--- Decode from Bellcore.

80 1C: Header (80 = Call Setup, Length)
01   : Parameter Value (Date and Time)
08   : Length of Information
31 31: Month (11 = November)
31 37: Day (17th)
32 32: Hour( 22)
30 35: Minute(05)
02   : Parameter Value (Calling Line DN)
07   : Length of Parameter
35 35 35 33 30 33 30 : Phone number (5553030)
07   : Parameter Value (Display)

!--- "P" (0x50) is sent if "Anonymous" indication
!--- is to be sent to phone.
!--- "O" (0x4F) is sent if "Out of Area/Unavailable" indication
!--- is to be sent to the phone.

07   : Parameter Length
4F 75 74 73 69 64 65 : Display in ASCII Hex.
```

Note: Lines of this output that are on more than one line are actually displayed as one line in the debug output.

In the example shown, everything works fine and both Name and Number Display are properly delivered to the phone. In these two scenarios, the calling number fails to display in one case and in the other case, the name fails to display.

Calling Number is Lost, Name is Delivered

```
Nov 17 17:39:34.164 EST: [1/1/0] htsp_set_caller_id_tx
   calling num= display_info=Outside called num=9913050
Nov 17 17:39:34.164 EST: [1/1/0] Caller ID String 80 16
   01 08 31 31 31 37 32 32 33 39 04 01 4F
```

```
      07 07 4F 75 74 73 69 64 65 88
```

Note: Lines of this output that are on more than one line are actually displayed as one line in the debug output.

When the hexadecimal Caller ID String is decoded in the example, the substring **04 01 4F** translates to these:

```
04 : Reason for Absence of DN
01 : Length of message
4F : "Out of Area"
```

Calling Number is Delivered, Name is Lost

```
Nov 17 17:53:24.034 EST: [1/1/0] htsp_set_caller_id_tx
   calling num=5551212 display_info= called num=9913050
Nov 17 17:53:24.034 EST: [1/1/0] Caller ID String 80 16
   01 08 31 31 31 37 32 32 35 33 02 07 35 35 35 31 32 31 32
   08 01 4F 05
```

Note: Lines of this output that are on more than one line are actually displayed as one line in the debug output.

When the hexadecimal Caller ID String is decoded in the example, the substring **08 01 4F** translates to these:

```
08 : Reason for Absence of Display
01 : Length
4F : "Out of Area"
```

These are the same VPM debugs for an FXO port which receives Caller ID. In the example shown, the FXS port transmits Caller ID to the phone. In the case of an FXO port, the process is reversed, but the debugs are very similar (shown here).

Debugs for an FXO Port Receiving Caller ID Correctly

```
Nov 20 10:40:15.861 EST: [1/0/0] htsp_start_caller_id_rx
Nov 20 10:40:15.861 EST: [1/0/0] htsp_set_caller_id_rx:BELLCORE
Nov 20 10:40:15.861 EST: htsp_timer − 10000 msec
Nov 20 10:40:17.757 EST: [1/0/0, FXOLS_RINGING, E_DSP_SIG_0100]
Nov 20 10:40:17.757 EST: fxols_ringing_not
Nov 20 10:40:17.761 EST: htsp_timer_stop
Nov 20 10:40:17.761 EST: htsp_timer − 10000 msec
Nov 20 10:40:18.925 EST: [1/0/0] htsp_stop_caller_id_rx
Nov 20 10:40:21.857 EST: [1/0/0, FXOLS_RINGING, E_DSP_SIG_0000]
Nov 20 10:40:23.857 EST: [1/0/0, FXOLS_RINGING, E_DSP_SIG_0100]
Nov 20 10:40:23.857 EST: fxols_ringing_not
Nov 20 10:40:23.861 EST: htsp_timer_stop htsp_setup_ind
Nov 20 10:40:23.861 EST: [1/0/0] get_fxo_caller_id:Caller ID received.
   Message type=128 length=31 checksum=74
Nov 20 10:40:23.861 EST: [1/0/0] Caller ID String 80 1C
   01 08 31 31 32 30 31 35 34 30 02 07 35 35 35 31 32 31 32
   07 07 4F 7574 73 69 64 65 74
Nov 20 10:40:23.865 EST: [1/0/0] get_fxo_caller_id
   calling num=5551212 calling name=Outside calling time=11/20 15:40
Nov 20 10:40:23.869 EST: [1/0/0, FXOLS_WAIT_SETUP_ACK, E_HTSP_SETUP_ACK]
Nov 20 10:40:23.873 EST: fxols_wait_setup_ack:
Nov 20 10:40:23.873 EST: [1/0/0] set signal state = 0xC timestamp = 0
Nov 20 10:40:23.985 EST: [1/0/0, FXOLS_PROCEEDING, E_DSP_SIG_0100]
   fxols_proceed_clear
Nov 20 10:40:23.985 EST: htsp_timer_stop2
Nov 20 10:40:24.097 EST: [1/0/0, FXOLS_PROCEEDING,E_DSP_SIG_0110]
   fxols_rvs_battery
Nov 20 10:40:24.097 EST: htsp_timer_stop2
```

```
Nov 20 10:40:24.733 EST: [1/0/0, FXOLS_PROCEED_RVS_BT,E_HTSP_PROCEEDING]
   fxols_offhook_proc
Nov 20 10:40:24.733 EST: htsp_timer − 120000 msec
Nov 20 10:40:24.745 EST: [1/0/0, FXOLS_PROCEED_RVS_BT,E_HTSP_VOICE_CUT_THROUGH]
   fxols_proc_voice
```

Note: Lines of this output that are on more than one line are actually displayed as one line in the debug output.

Related Information

- **Caller ID on Cisco 2600 and 3600 Series Routers and Cisco MC3810 Multiservice Concentrators**
- **Voice Technology Support**
- **Voice and Unified Communications Product Support**
- **Troubleshooting Cisco IP Telephony**
- **Technical Support − Cisco Systems**

Contacts & Feedback | Help | Site Map
© 2013 − 2014 Cisco Systems, Inc. All rights reserved. Terms & Conditions | Privacy Statement | Cookie Policy | Trademarks of Cisco Systems, Inc.

Updated: Jul 02, 2007 Document ID: 23444