

Troubleshooting Mute Calls on the Cisco PGW 2200

Document ID: 44183

Introduction

Before You Begin

- Conventions

- Prerequisites

- Components Used

Call Flows

- Typical Call Flow with Cut-Through on ANM

- Typical Call Flow with Cut-Through on ACM

Customers Report Muted Calls

- Collecting Additional Call Information

Possible Causes and Recommended Actions

- Understanding and Defining Your Problem

- Checking CPU Load on PGW 2200

- Checking CPU Load on the Gateways

- RSIP Messages Not Sent on Secondary Ethernet

- Mute Call Troubleshooting Suggestions

- Additional Tasks Related to Mute Calls

- Understanding the Error Log

- Transaction Deleted

Collecting an MDL Trace on the PGW 2200

Related Information

Introduction

This document provides troubleshooting information for voice calls that are muted in one direction on the Cisco PGW 2200 PSTN Gateway (Cisco PGW 2200). The information in this document applies specifically to the Cisco PSTN Gateway Solution using Media Gateway Controller (MGC) and Cisco AS5x00 gateways in combination with the Cisco PGW 2200.

Before You Begin

Conventions

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

Prerequisites

There are no specific prerequisites for this document.

Components Used

The information in this document is based on the software and hardware versions below.

Platform	Platform name	Release
----------	---------------	---------

PGW 2200 Node	Cisco MGC	9.2(2) [From patch S(29)] 9.3(2) [From patch S(7)] 9.4(1)
PSTN Gateway	AS5x00	12.2T or higher

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

Call Flows

Understanding the different settings on cut-through via the routing.mml can help you understand the different call flows for the Cisco PGW 2200 concept.

Typical Call Flow with Cut-Through on ANM

A typical call flow with cut-through on ANM (such as 3) is shown below:

```

Originating TDM Originating Gateway PGW Terminating Gateway Terminating TDM
-----IAM----->
<-CRCX--
(M: inactive)
--- OK---->

<-----ACM-----
<-MDCX--
(M: recvonly)
--OK----->

<-----ANM-----
<--MDCX---
(M: sendrecv)
----OK---->

```

Note: The optional MDCX is sent to the Terminating Gateway to turn on echo cancellation only if:

- the trunk group property "EchoCanRequired" is set, and
- the Term TDM switch did not provide echo cancellation (for example, the Echo_control_device_indicator parameter in ACM message from Term TDM was set to zero).

Typical Call Flow with Cut-Through on ACM

A typical call flow with cut-through on ACM (such as cut-through equals 2) is shown below:

```

Originating TDM Originating Gateway PGW Terminating Gateway
-----IAM----->
<--CRCX---
(M: inactive)

```

--- OK----->

(M:

```
<-----ACM-----  
      <---MDCX---  
      (M: sendrecv)  
      ---OK--->
```

```
<-----ANM-----
```

Note: When the call is busy or there is some sort of announcement to play to the caller, there is no reason to open the voice path in both directions. If you think you have a reported mute call, issue a **debug mgcp packet** command on the Originating and Terminating Gateway in combination with a **show call history voice** command linked to the messages with a zero sent packet count, a **show call history voice brief** command, and a Cisco Message Definition Language (MDL) trace on the PGW 2200 to understand the problem. A snooper trace will also help you to understand the problem. The MDL trace provides a complete SS7 and Media Gateway Control Protocol (MGCP) call flow.

Customers Report Muted Calls

The following conditions cause the PGW 2200 to flag mute calls (during Delete Connection [DLCX]) and detection in platform.log. These logs contain a call ID which has the gateway information and CIC information.

1. PGW 2200 is configured in Fault Tolerant mode.
2. Call was answered (the call was successfully cut-through.).
3. 250 OK message was received with (P:) in response to DLCX.
4. Either Packet Sent (PS) equals 0 or Packet Received (PR) equals 0 in (P:).
5. Call duration was more than 1 second.

Collecting Additional Call Information

To collect additional call information, use the following steps:

1. Make a Telnet connection to gateway **as5xxx-1**.
2. Issue the following command to find the call ID back again which is linked with the endpoint and the mute call message from platform.log:

```
as5xxx-1 > show mgcp connection
```

The following is sample output from the **show mgcp connection** command for Voice over IP (VoIP) connections:

```
Endpoint Call_ID(C) Conn_ID(I) (P)ort (M)ode (S)tate (C)odec (E)vent[SIFL] (R)esult  
1. S0/DS1-0/1 C=103,23,24 I=0x8 P=16586,16634 M=3 S=4,4 C=5 E=2,0,0,2 R=0,0
```

Field descriptions of the **show mgcp connection** command for VoIP are shown below.

- **Endpoint** The endpoint for each call, shown in the digital endpoint naming convention of slot number (S0) and digital line (DS1-0) number (1).

- **Call_ID(C)** The MGCP call ID sent by the call agent, the internal Call Control Application Programming Interface (CCAPI) call ID for this endpoint, and the CCAPI call ID for the peer call legs. CCAPI is an API that provides call control facilities to applications.
- **Conn_ID(I)** The connection ID generated by the gateway and sent in the ACK message.
- **(P)ort** The ports used for this connection. The first port is the local User Datagram Protocol (UDP) port. The second port is the remote UDP port.
- **(M)ode** The call mode, where:
 - ◆ 0 Indicates an invalid value for mode.
 - ◆ 1 Indicates that the gateway should only send packets.
 - ◆ 2 Indicates that the gateway should only receive packets.
 - ◆ 3 Indicates that the gateway can send and receive packets.
 - ◆ 4 Indicates that the gateway should neither send nor receive packets. [Inactive]
 - ◆ 5 Indicates that the gateway should place the circuit in loopback mode.
 - ◆ 6 Indicates that the gateway should place the circuit in test mode.
 - ◆ 7 Indicates that the gateway should use the circuit for network access for data.
 - ◆ 8 Indicates that the gateway should place the connection in network loopback mode.
 - ◆ 9 Indicates that the gateway should place the connection in network continuity test mode.
 - ◆ 10 Indicates that the gateway should place the connection in conference mode.
- **(S)tate INFO** Example: S=4,4 The first integer shows the local MGCP call state and the second integer shows the remote MGCP call state.

MGCP_CALL_IDLE	= 0, Idle
MGCP_CALL_SETTING	= 1, Incoming call from PSTN
MGCP_CALL_CONNECTING	= 2, MGCP CRCX received
MGCP_CALL_CONFERENCING	= 3, Call connected, await conf
MGCP_CALL_ACTIVE	= 4, Conference created
MGCP_CALL_CONF_DESTROYING	= 5, Destroying conference
MGCP_CALL_DISCONNECTING	= 6, Conf destroyed, disconnect call
MGCP_CALL_INACTIVE	= 7, Call in inactive mode
MGCP_CALL_VOICE_CONNECTING	= 8, Creating telephony call leg only
MGCP_CALL_VOICE_ACTIVE	= 9, Telephony

	call leg created
MGCP_CALL_CONF DISSOCIATING	= 10, Destroying conf
MGCP_CALL_CALLLEGS DISSOCIATED	=11, Conf destroyed, no call discon
MGCP_CALL_HP_CONNECTING	= 12, Connecting TDM hairpin call leg
MGCP_CALL_HP_CONNECTED	= 13, One HP call leg connected
MGCP_CALL_HP_CONFERENCING	= 14, Conferencing TDM Hairpin call leg
MGCP_CALL_HP_ACTIVE	= 15, TDM hairpining active state
MGCP_CALL_VOIP_CONF_DESTROY	= 16, Conf destroyed, make HP call
MGCP_CALL_ERROR_STATE	= 17, Call in error state
MGCP_CALL_CONNECTING_INACTIVE	= 18, Creating inactive connection
MGCP_CALL_CONF_DESTROYING_INACTIVE	= 19, Conf destroy inactive conn
MGCP_CALL_CONT_TEST	= 20, AAL2/IP continuity test (xrbk)
MGCP_CALL_SETUP_WAIT	= 21, Waiting for setup information
MGCP_CALL_WAIT_NSE_SENT	= 22, Wait for NSE event to be sent
MGCP_CALL_TWC_ACTIVE	= 23, TWC call active
MGCP_CALL_WAIT_STATE	= 24, App is

	waiting for call control
MGCP_CALL_HANDOVER	= 25, App is grabbing back the control
MGCP_CALL_EM_DISCONNECTING	= 26, Disconnect call, E&M endpoints
MGCP_CALL_MAX_STATE	= 27

• (C)odec INFO Example: C=1

MGCP_CODEC_UNDEFINED	0
MGCP_G711_U,	1 = G.711 mu-law
MGCP_G711_A,	2 = G.711 A-law
MGCP_G726_32K,	3 = G.726 32K
MGCP_G726_24K,	4 = G.726 24K
MGCP_G726_16K,	5 = G.726 16K
MGCP_G729,	6 = G.729
MGCP_G729_A,	7 = G.729 A
MGCP_G729_B,	8 = G.729 B
MGCP_G729_B_LOW_COMPLEXITY,	9 = G.729 B
MGCP_G728,	10 = G.728
MGCP_G7231_HIGH_RATE,	11 = G.723.1 High Rate
MGCP_G7231_A_HIGH_RATE,	12 = G.723.1 Annex A High Rate
MGCP_G7231_LOW_RATE,	13 = G.723.1 Low Rate
MGCP_G7231_A_LOW_RATE,	14 = G.723.1 Annex A Low Rate
MGCP_GSM_FULL_RATE,	15 = GSM full rate
MGCP_GSM_HALF_RATE,	16 = GSM half rate
MGCP_GSM_ENHANCED_FULL_RATE,	17 = GSM

	enhanced full rate
MGCP_GSM_ENHANCED_HALF_RATE,	18 = GSM enhanced half rate
MGCP_CLEAR_CHANNEL = 128,	128 = Nx64 clear channel
MGCP_NSE = 129	For NSE

- **(E)vent** Example: E=3,0,2,3 The event field is read as: E=last_successful_mgcp_event, last_successful_internal_event, last_failed_app_event, last_requested_app_event

MGCP_APP_EV_ACK = -1,	MGCP ACK
MGCP_APP_EV_CREATE_CONN = 0,	MGCP create connection msg
MGCP_APP_EV_DELETE_CONN, =1	MGCP delete connection msg
MGCP_APP_EV_MODIFY_CONN, =2	MGCP modify connection msg
MGCP_APP_EV_NOTIFY_REQ, =3	MGCP notify request msg
MGCP_APP_EV_ALERT, =4	CCAPI alert event
MGCP_APP_EV_CALL_CONNECT,=5	CCAPI call connect event
MGCP_APP_EV_CONF_RDY,=6	CCAPI conference ready
MGCP_APP_EV_CONF_DESTROY,=7	CCAPI conference destroyed
MGCP_APP_EV_CALL_DISCONNECT,=8	CCAPI call disconnect
MGCP_APP_EV_CALL_PROCEED, =9	CCAPI call proceeding
MGCP_APP_EV_OFF_HOOK,=10	CCAPI off-hook/call setup ind
MGCP_APP_EV_ON_HOOK, =11	CCAPI on-hook/call disconnected
MGCP_APP_EV_MEDIA_EVT,=12	MGCP Media

	Events
MGCP_APP_EV_INT_EVT,=13	MGCP Internal Events
MGCP_APP_EV DISSOC_CONF,=14	
MGCP_APP_EV_ASSOC_CONF,=15	
MGCP_APP_EV_MODIFY_DONE,=16	CCAPI Call modify done ev
MGCP_APP_EV_VOICE_MODE_DONE,=17	Voice Cut-thru has happened
MGCP_APP_EV_NSE,=18	CCAPI NSE events
MGCP_APP_EV_CALL_HANDOFF,=19	Handoff Call to some other app
MGCP_APP_EV_MAX_EVENT	

- **(R)result** Example: R=0,0 The result field is interpreted as: R=Event_result,(boolean) do we need to send ACK?

MGCP_APP_EVR_NORMAL_OK = 0,	MGCP normal event processed OK
MGCP_APP_EVR_INVALID_OK,	MGCP invalid event processed OK
MGCP_APP_EVR_CALLP_REL,	the call record is released
/* MGCP Protocol errors */	
MGCP_APP_EVR_INVALID_CALL_ID = 10,	TGW find invalid call id
MGCP_APP_EVR_INVALID_CONNECTION_ID,	TGW find invalid connection id
MGCP_APP_EVR_DUPLICATED_MESSAGE,	TGW find duplicated sgcp msg
MGCP_APP_EVR_MGCP_ACK_FAILURE,	TGW cannot send sgcp ack msg
MGCP_APP_EVR_MGCP_DELETE_FAILURE,	TGW cannot send sgcp delete msg
MGCP_APP_EVR_MGCP_CREATE_ACK_FAILURE,	

	TGW cannot send create ack msg
MGCP_APP_EVR_MGCP_CREATE_ACK_MISSING,	TGW did not send sgcp ack msg
MGCP_APP_EVR_MGCP_DELETE_ACK_FAILURE,	TGW cannot send delete ack msg
MGCP_APP_EVR_MGCP_NOTIFY_FAILURE,	TGW cannot send sgcp notify msg
MGCP_APP_EVR_INVALID_STATE,	TGW find event in wrong state
/* Resource problem */	
MGCP_APP_EVR_TGW_DOWN = 30,	TGW in graceful shutdown mode
MGCP_APP_EVR_TGW_NOT_READY,	TGW not ready for the event
MGCP_APP_EVR_CALL_VDB_FAILURE,	TGW cannot obtain the vdbptr
MGCP_APP_EVR_PREV_RTP_PORT_LOCKED,	TGW find previous rtp port locked
MGCP_APP_EVT_CONN_RECORD_MISSING,	TGW cannot find conn record
MGCP_APP_EVR_ENDP_NOT_READY,	TGW not ready for the event
MGCP_APP_EVR_MEM_RSRC_ERROR,	TGW has transient mem alloc err
MGCP_APP_EVR_CALL_CAC_FAILURE,	GW does not have the bandwidth
MGCP_APP_EVR_CONF_RSRC_ERROR,	GW cannot get conf resource
/* Event failure */	
MGCP_APP_EVR_REQ_EVENT_FAILURE = 40,	TGW cannot handle requested event
MGCP_APP_EVR_INVALID_CCAPI_EVENT,	TGW cannot handle the ccapi event

MGCP_APP_EVR_IGNORE_CCAPI_EVENT,	TGW will ignore the ccapi event
/* Signal failure */	
MGCP_APP_EVR_SIGNAL_FAILURE = 50,	TGW cannot handle the signal
MGCP_APP_EVR_ABNORMAL_ONHOOK,	TGW find abnormal onhook
MGCP_APP_EVR_INVALID_OFFHOOK,	TGW find invalid offhook
MGCP_APP_EVR_INVALID_COT,	TGW find invalid cot
MGCP_APP_EVR_COT_FAILURE,	TGW failed to do COT
MGCP_APP_EVR_COT_DISABLE_FAILURE,	TGW failed to disable COT
/* Call setup failure */	
MGCP_APP_EVR_CALL_SETUP_REQ_FAILURE = 60,	TGW cannot setup call request
MGCP_APP_EVR_CALL_SETUP_IND_FAILURE,	TGW cannot handle call indication
MGCP_APP_EVR_CALL_CONTEXT_FAILURE,	TGW cannot setup the context
MGCP_APP_EVR_CALL_PEER_FAILURE,	TGW cannot setup the peer
MGCP_APP_EVR_CALL_VOX_CALL_FAILURE,	TGW cannot setup the voip/voaal2 call
MGCP_APP_EVR_CALL_VOIP_CALL_FAILURE,	TGW cannot setup the voip call
MGCP_APP_EVR_CALL_DISCONNECT_FAILURE,	TGW cannot disconnect the call
MGCP_APP_EVR_CALL_MODIFY_FAILURE,	TGW cannot modify the call parm
MGCP_APP_EVR_CALL_ALERT_FAILURE,	TGW cannot alert the call
MGCP_APP_EVR_CALL_DELETE_FAILURE,	

	TGW cannot delete the call
MGCP_APP_EVR_CALL_UNKNOWN_FEATURE,	TGW cannot handle unknow feature
MGCP_APP_EVR_UNSUPPORTED_CODEC,	TGW find unsupported codec
MGCP_APP_EVR_NO_DIGIT_MAP,	TGW cannot find the digit map
MGCP_APP_EVR_IGNORE_DIGIT,	TGW cannot process the digits
MGCP_APP_EVR_DIGITS_OVERFLOW,	TGW cannot handle too many digits
MGCP_APP_EVR_DIGITS_NOTIFY_FAILURE,	TGW cannot send the digits out
MGCP_APP_EVR_CODEC_NOT_MATCHED,	TGW codec doesn't match rmt TGW
MGCP_APP_EVR_INVALID_CONN_MODE,	TGW cannot understand con mode
/* Peer failure */	
MGCP_APP_EVR_PEER_MISSING = 90,	TGW find not find the peer
MGCP_APP_EVR_PEER_NOT_READY,	TGW find peer not ready
MGCP_APP_EVR_PEER_IN_WRONG_STATE,	TGW find the peer in wrong state
MGCP_APP_EVR_PEER_DISCONNECT_FAILURE,	TGW cannot disconnect the peer
MGCP_APP_EVR_NO_CONFERENCE_ID,	TGW cannot find the conference ID
MGCP_APP_EVR_CONF_CREATE_FAILURE,	TGW cannot create conference
MGCP_APP_EVR_CONF_DESTROY_FAILURE,	TGW cannot destroy conference

MGCP_APP_EVR_UNKNOWN_CONN_TYPE,	TGW cannot handle the con type
MGCP_APP_EVR_INVALID_ENDPOINT,	TGW cannot connect to endpoint
MGCP_APP_EVR_INVALID_NSE_EVENT = 100,	Invalid NSE event
MGCP_APP_EVR_NSE_RCVD_ON_WRONG_LEG,	The NSE events come to a wrong leg
MGCP_APP_EVR_SEND_NSE_FAILURE,	Cannot send an NSE event
MGCP_APP_EVR_PLAY_TONE_FAILURE,	Cannot play NSE-requested tone
/* Transient Error */	
MGCP_APP_EVR_TRANS_ERROR = 110,	TGW endpoint in transition state
MGCP_APP_EVR_MAX_RESULT	

Possible Causes and Recommended Actions

Understanding and Defining Your Problem

Mute calls may be linked to software problems or other issues. Use the following steps to begin to troubleshoot mute calls on the Cisco PGW 2200.

1. Understand the customer's problem description. Mute calls can be related to other items which are not linked to software errors such as IP routing and Layer 1 problems. The solution of each root cause often exposes additional lower-level problems that need to be fixed first.
2. Calculate the ratio of failed calls to mute calls at customer location per 24-hour monitoring.
3. Avoid defining exactly what percentage of calls are alarming.
4. Try to reproduce this situation to understand the real cause of the problem.

Checking CPU Load on PGW 2200

To check CPU load on the PGW 2200, issue the following command:

```
mml> rtrv-ne-health
```

This command displays the following type of information:

```
MGC-01 - Media Gateway Controller 2003-02-14 15:36:50.788 GMT
M RTRV
"Platform State:ACTIVE"
"Machine Congestion Level = MCL 0 (No Congestion)"
"Current in progress calls = 83, call attempts = 2 cps"
"CPU 0 Utilization = 1 % CPU 1 Utilization = 0 %"
```

```
"CPU 2 Utilization = 2 % CPU 3 Utilization = 0 %"  
"Memory (KB):3715344 Free virtual, 8390328 Total virtual, 4194304  
Total rea"  
"Filesystem kbytes used avail capacity Mounted on"  
"/dev/dsk/c0t0d0s0 494235 47099 397713 11% /"  
"/dev/dsk/c0t0d0s4 10678328 5494165 5077380 52% /opt"  
;
```

Check the Calls Per Second (CPS) information and try to calculate this in combination with the gateways you are using. Maybe some gateways have a high CPU load due to the amount of calls coming in. The following result display into platform.log also can explain the status of the system.

```
Fri Nov 13 10:18:28:119 2002 CET | engine (PID 14488) <Error>engMclMgrImpl::updateSystemMc  
System Mcl = 1, MclName = cpu, Load = 84 AvgLoad = 68
```

Note: In this example, there was a CPU overload condition due to high traffic which abated in a few minutes. This is as a result of peak hour period. At that point, try to link this information with mute calls.

Checking CPU Load on the Gateways

To obtain a status over a certain amount of time, issue the following command:

```
as5xxx-1> show proc cpu history
```

A high CPU load can be caused by one of the items being linked to process switching. To check this, issue the **show running-config | incl route** command.

```
as5xxx-1> show running-config | incl route
```

To avoid a high CPU load on the gateway, do *not* have the following commands in your configurations:

```
no ip route-cache  
no ip route-cache cef
```

Note: The **ip route-cache** or **ip route-cache cef** command should be configured on the gateways.

If you see any of the above, you are most likely process switching instead of fast switching and the system load will be extremely high, calls may be lost, and the voice quality will be poor. Additionally, the MGCP message may not be Acknowledged (ACK) or generated.

RSIP Messages Not Sent on Secondary Ethernet

Depending on the way the **ip host** command is configured on the gateways, it will not send RSIP messages on the secondary Ethernet. The reason the gateway is trying to send to the first IP address for a second round of tries before failing over to the second IP address is linked to the Cisco IOS® Software configuration. This forces a DNS lookup (which looks at an **ip host** command when **no ip domain-lookup** command is configured). When this happens, the first IP address is returned and used again. To avoid this behavior, use the following command in the MGCP profile:

```
as5xxx-1> mgcp profile  
as5xxx-1> no max1 lookup
```

Note: You need to reload the router for the **no max1 lookup** command configuration to take effect.

Mute Call Troubleshooting Suggestions

Perform the following steps to determine if there are additional issues in your network.

1. Determine if call duration is less than 10 seconds.
2. Determine if the Transmit (Tx) packets or Receive (Rx) packets are zero.

```
as5xxx-1> show mgcp connection
```

And check on Call_ID, which is 3334373 for this example.

```
Endpoint                Call_ID(C)  Conn_ID(I)                                (P)ort
1. S6/DS1-1/31 C=345F3D,3334373,3334374  I=0x197074  P=19544,18424  M=3
```

3. Try to link Call_ID using the following:

```
as5xxx-1 > show call active voice brief | incl Call_ID
```

```
Tele 0/0:0 (call_id): tx:0/0/0ms None noise:0 acom:0 i/0:0/0 dBm
```

4. At this point, you can find the information from the **show call active voice** command, linked on Conn_ID to find Tx packets, Tx bytes, Rx packets, and Rx bytes information. This information can tell you the numbers of packets being sent and received.

```
Telephony call-legs: 1
SIP call-legs: 0
H323 call-legs: 0
Total call-legs: 2
0 : 482619719hs.1 +0 pid:0 Originate active
dur 00:12:35 tx:42517/711257 rx:24197/661142
Tele 6/1:0 (3334373): tx:755060/278000/0ms g729r8 noise:-120 acom:90 i/0:-51/-12
0 : 482619719hs.2 +-1 pid:0 Originate connecting
dur 00:00:00 tx:24192/660942 rx:42517/711257
IP 0.0.0.0:18424 rtt:1ms pl:280000/37390ms lost:347/1/0 delay:40/30/120ms g729r8
```

In this case, you can find the Local and Remote Gateway details.

```
as5xxx-1 > show voip rtp connections
```

```
VoIP RTP active connections :
```

```
No. CallId dstCallId LocalRTP RmtRTP LocalIP RemoteIP
1 3334374 3334373 19544 18424 193.41.31.2 193.41.24.5
```

5. Determine if a larger percentage of the mute calls occur during busy periods.

In rare situations, packets sent by the Cisco AS5400 may not be received by the Cisco AS5300 TDM interface. If this occurs, the Cisco AS5400 DLCX ACK shows Tx packets, but the Cisco AS5300 does not show any Rx packets. Loopback interface is important for the MGCP connection in combination with the **mgcp bind** command.

Note: MGCP implementation uses the best available IP address on the MGC as a source address to communicate with the call agent. The media stream uses the loopback address if configured, otherwise the best available IP address as its source address. There is no defined way to change this behavior. The **bind** command allows more flexibility for choosing source address for both control and media packets.

Listed below are a few situations which explain the behavior of the command. For all of these cases, appropriate warning message will be displayed depending on the situation.

- When there are active MGCP calls on the gateway, the **bind** command will be rejected for both control and media.
- If the bind interface is not up, then the command is accepted, but it does not take affect until interface

comes up.

- If the IP address is not assigned on the bind interface, the **bind** command is accepted but it takes effect only after valid IP address is assigned, during this time if MGCP calls are up, the **bind** command is removed.
- When the bound interface goes down, either because of a manual shut on the interface or because of an operational failure, bind activity is disabled on that interface.
- When bind is not configured on the MGC, the IP address used for sourcing MGCP control and media is best available IP address.

Additional Tasks Related to Mute Calls

One of the criteria that the PGW 2200 uses to flag a mute call is that the call should be in Answer state, which means that the ANM message should have been sent by the PGW 2200 to the originating SS7 switch. Before sending the ANM message to the originating SS7 switch, the PGW 2200 transmits the MDCX to the GW to set the Mode to Send–Recv. If the MDCX is not acknowledged by the GW due to connectivity or other issues, the call does not reach the Answer state, hence it is not tracked as a mute call. At that point, an Error log will be sent to the platform.log file at opt/CiscoMGC/var/log.

MGCP Command Resent

If MGCP command message (CRCX, DLCX, MDCX) is resent due to a time–out (for example PGW 2200 sent MDCX [sendrecv] four times), but the gateway did not ACK it, the call fails and it is not considered a mute call by the PGW 2200. PGW 2200 flags a mute call (mute–message into platform.log) during DLCX if:

1. The call was answered, and
2. a 250 OK message had connection parameter (P:), and
3. Either PS or PR was 0 in (P:)

Note: This can be linked to other items not linked as a real mute call. For example, if the calling party hangs up when the called party answers, you see this message and it is correct. But this is not a mute call. For hairpin calls (hairpinning is the name given to calls that originate and terminate on the same router or gateway), the 250 OK message in response to DLCX does not have connection parameter (P:). These calls are not flagged as mute.

Understanding the Error Log

An error is written in the following format for re–sending the information:

```
mgcp_link_comp_id ioCcMgcpConnMgr: mgcpCmdRequestTimeout: Successfully resent txn:transact.  
message cnt:no_of_retry remaning
```

Example:

```
Tue Jul 16 11:05:46:219 2002 EST | mgcp-1 (PID 20828) <Error>  
00100001 ioCcMgcpConnMgr: mgcpCmdRequestTimeout: Successfully resent txn:1718 msg:DLCX 171  
C: 72  
I: 16  
R:  
S:  
X: 6B5  
cnt:1.
```

Transaction Deleted

If a transaction is deleted after maximum retries, an error is written in the following format:

```
MGCP Link Comp Id ioCcMgcpConnMgr: mgcpCmdRequestTimeout: type message type, cnt: <-1>,
  txn:transaction_id, connMsgPtr pointer to message
```

Example:

```
Tue Jul 16 11:05:50:218 2002 EST | mgcp-1 (PID 20828) <Error>
00100001 ioCcMgcpConnMgr: mgcpCmdRequestTimeout: type 5, cnt:-1, txn: 1718, connMsgPtr 002
```

Issue the **show mgcp stat** command to check on failed items and try to understand why a transaction was deleted.

Collecting an MDL Trace on the PGW 2200

If all items are correct, run an MDL trace and collect all of the details from the **show log** command on the GW. The following steps show how to collect MDL trace:

1. Identify the Originating SS7 SigPath Number or Originating TrunkGroup Number on which calls are placed.
2. Start the MDL trace by issuing the following command:

```
mml> sta-sc-trc:ss7sigPath name | orig
      trunkgroup number
```

3. Perform a test.
4. Stop the MDL trace by issuing the following command:

```
mml> stp-sc-trc:all
```

5. Identify the Call Id (C:) of the bad call from MGCP debug on the gateway.
6. Convert the MDL trace into a readable format:

```
mml> get_trc.sh trace file name
```

7. Type **Call Id** at the prompt to jump to the MDL trace of the bad call.
8. Choose **option C** to convert the trace file.
9. The trace file is in **/opt/CiscoMGC/var/trace**.

Related Information

- [Cisco PGW 2200 Softswitch Tech Notes](#)
 - [Recommended Reading: Troubleshooting Cisco IP Telephony](#)
 - [Technical Support – Cisco Systems](#)
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Feb 02, 2006

Document ID: 44183
