

Guide Contents
Troubleshooting Cisco IOS Voice Overview
Debug Command Output on Cisco IOS Voice Gateways
Filtering Troubleshooting Output
Cisco VoIP Internal Error Codes
Troubleshooting Cisco IOS Voice Telephony
Troubleshooting Cisco IOS Voice Protocols
Troubleshooting Cisco IOS Telephony Applications
Monitoring the Cisco IOS Voice Network
Cause Codes and Debug Values

To troubleshoot xGCP call routing, see the following sections:

- [Verifying Digits Received and Sent on the POTS Call Leg](#)
- [Verifying End-to-End VoIP Signaling on the VoIP Call Leg](#)

Contents

- [1 Verifying Digits Received and Sent on the POTS Call Leg](#)
 - ◆ [1.1 show dialplan number](#)
 - ◆ [1.2 debug vtsp dsp](#)
- [2 Verifying End-to-End VoIP Signaling on the VoIP Call Leg](#)

Verifying Digits Received and Sent on the POTS Call Leg

Once the on-hook and off-hook signaling are verified to be working correctly, the next step in troubleshooting and debugging a VoIP call is to verify that the correct digits are being received or sent on the voice port (digital or analog). A dial peer is not matched or the switch (CO or PBX) cannot ring the correct station if incomplete or incorrect digits are being sent or received. Some commands that can be used to verify the digits received/sent are:

- **show dialplan number**-This command is used to show which dial peer is reached when a particular telephone number is dialed.
- **debug vtsp session**-This command displays information on how each network indication and application request is processed, signaling indications, and DSP control messages.
- **debug vtsp dsp** -This command displays the digits as they are received by the voice port.
- **debug vtsp all**-This command enables the following debug voice telephony service provider (VTSP) commands: **debug vtsp session**, **debug vtsp error**, and **debug vtsp dsp**.

show dialplan number

The **show dialplan number** *digit_string* command displays the dial peer that is matched by a string of digits. If multiple dial peers can be matched, they are all shown in the order in which they are matched. The output of this command looks like this:

```
Router# show dialplan number 5000
Macro Exp.: 5000
VoiceOverIpPeer2
    information type = voice,
    tag = 2, destination-pattern = `5000',
```

Cisco_IOS_Voice_Troubleshooting_and_Monitoring_--_MGCP_Call_Routing_and_Dial_Peers

```
answer-address = `', preference=0,
group = 2, Admin state is up, Operation
state is up,
incoming called-number = `',
connections/maximum = 0/unlimited,
application associated:
type = voip, session-target =
`ipv4:192.168.10.2',
technology prefix:
ip precedence = 5, UDP checksum =
disabled, session-protocol = cisco,
req-qos = best-effort,
acc-qos = best-effort,
dtmf-relay = cisco-rtp,
fax-rate = voice,
payload size = 20 bytes
codec = g729r8,
payload size = 20 bytes,
Expect factor = 10, Icpif = 30,
signaling-type = cas,
VAD = enabled, Poor QOV Trap = disabled,
Connect Time = 25630, Charged Units = 0,
Successful Calls = 25, Failed Calls = 0,
Accepted Calls = 25, Refused Calls = 0,
Last Disconnect Cause is "10 ",
Last Disconnect Text is "normal call
clearing.",
Last Setup Time = 84427934.
Matched: 5000 Digits: 4
Target: ipv4:192.168.10.2
```

debug vtsp dsp

debug vtsp dsp shows the digits as they are received by the voice port. The following output shows the collection of DTMF digits from the DSP:

```
Router# debug vtsp dsp
Voice telephony call control dsp debugging is on
!-- ACTION: Caller picked up handset and dialed
!-- digits 5000.
!-- The DSP detects DTMF digits. Digit 5 was
!-- detected with ON time of 130msec.
*Mar 10 17:57:08.505: vtsp_process_dsp_message:
MSG_TX_DTMF_DIGIT_BEGIN: digit=5,
*Mar 10 17:57:08.585: vtsp_process_dsp_message:
MSG_TX_DTMF_DIGIT_OFF: digit=5,
duration=130
*Mar 10 17:57:09.385: vtsp_process_dsp_message:
MSG_TX_DTMF_DIGIT_BEGIN: digit=0
*Mar 10 17:57:09.485: vtsp_process_dsp_message:
MSG_TX_DTMF_DIGIT_OFF: digit=0,
duration=150
*Mar 10 17:57:10.697: vtsp_process_dsp_message:
MSG_TX_DTMF_DIGIT_BEGIN: digit=0
*Mar 10 17:57:10.825: vtsp_process_dsp_message:
MSG_TX_DTMF_DIGIT_OFF: digit=0,
duration=180
*Mar 10 17:57:12.865: vtsp_process_dsp_message:
MSG_TX_DTMF_DIGIT_BEGIN: digit=0
*Mar 10 17:57:12.917: vtsp_process_dsp_message:
```

show dialplan number

```
MSG_TX_DTMF_DIGIT_OFF: digit=0,
duration=100
Router# debug vtsp session
Voice telephony call control session debugging is on
!-- <some output have been omitted>
!-- ACTION: Caller picked up handset.
!-- The DSP is allocated, jitter buffers, VAD
!-- thresholds, and signal levels are set.
*Mar 10 18:14:22.865: dsp_set_playout: [1/0/0 (69)]
packet_len=18 channel_id=1 packet_id=76 mode=1
initial=60 min=4 max=200 fax_nom=300
*Mar 10 18:14:22.865: dsp_echo_canceller_control:
[1/0/0 (69)] packet_len=10 channel_id=1 packet_id=66
flags=0x0
*Mar 10 18:14:22.865: dsp_set_gains: [1/0/0 (69)]
packet_len=12 channel_id=1 packet_id=91
in_gain=0 out_gain=65506
*Mar 10 18:14:22.865: dsp_vad_enable: [1/0/0 (69)]
packet_len=10 channel_id=1 packet_id=78
thresh=-38act_setup_ind_ack
*Mar 10 18:14:22.869: dsp_voice_mode: [1/0/0 (69)]
packet_len=24 channel_id=1 packet_id=73 coding_type=1
voice_field_size=80
VAD_flag=0 echo_length=64 comfort_noise=1
inband_detect=1 digit_relay=2
AGC_flag=0act_setup_ind_ack(): dsp_dtmf_mod
e()act_setup_ind_ack: passthru_mode = 0,
no_auto_switchover = 0dsp_dtmf_mode
(VTSP_TONE_DTMF_MODE)
!-- The DSP is put into "voice mode" and dial-tone is
!-- generated.
*Mar 10 18:14:22.873: dsp_cp_tone_on: [1/0/0 (69)]
packet_len=30 channel_id=1 packet_id=72 tone_id=4
n_freq=2 freq_of_first=350 freq_of_second=440 amp_of_first=
4000 amp_of_second=4000 direction=1 on_time_first=65535
off_time_first=0 on_time
_second=65535 off_time_second=0
```

If you determine that the digits are not being sent or received properly, then you might need to use either a digit-grabber (test tool) or T1 tester to verify that the digits are being sent at the correct frequency and timing interval. If they are being sent "incorrectly" for the switch (CO or PBX), some values on the router or switch (CO or PBX) might need to be adjusted so that they match and the devices can interoperate. These are usually digit duration and interdigit duration values. If the digits appear to be sent correctly, you can also check any number translation tables in the switch (CO or PBX) that might be adding or removing digits.

Verifying End-to-End VoIP Signaling on the VoIP Call Leg

After verifying that voice-port signaling is working properly and that the correct digits have been received, move to the VoIP call control troubleshooting and debugging. The following factors explain why call control debugging can be a complex job:

- H.323 is made up of three layers of call-negotiation and call-establishment: H.225, H.245, and H.323. These protocols use a combination of TCP and UDP to set up and establish a call.
- End-to-End VoIP debugging shows a number of Cisco IOS state-machines, and problems with any state-machine can cause a call to fail.
- End-to-End VoIP debugging can be very verbose and create a lot of debug output.