



Configuring MGCP Gateway Support

This section provides information on configuring the MGCP Gateway Support for the **mgcp bind** command feature.

Feature benefits include the following:

- Media gateway controller-to-media gateway (MGC-to-MG) signaling and identification

The command allows you to use a loopback interface IP address for sourcing MGCP packets, which is transparent to any interface failure.

- Security of the media gateway

The command allows you to obtain a predefined interface for both MGCP and media control, which can be used for security configuration.

- Possible clash of voice and dial addressing

This feature allows you to assign a media bind interface other than loopback 0, which allows dial calls to conserve IP addresses.

- No interface diversity using routing and reduced MGCP voice diversity

You can use routing capability more efficiently if you configure the loopback interface for control. Using the command to configure the loopback interface helps in creating redundant MGCP control or media interface.

- MGCP backward compatibility

This feature is backward compatible with earlier MGCP features.

For more information about this and related Cisco IOS voice features, see the following:

- "Overview of MGCP and Related Protocols" on page 3
- Entire Cisco IOS Voice Configuration Library--including library preface and glossary, other feature documents, and troubleshooting documentation--at http://www.cisco.com/en/US/docs/ios/12_3/vvf_c/cisco_ios_voice_configuration_library_glossary/vcl.htm

Feature History for MGCP Gateway Support for the mgcp bind Command

Release	Modification
12.2(13)T	This feature was introduced.

- [Finding Feature Information, on page 2](#)
- [Prerequisites for Configuring MGCP Gateway Support, on page 2](#)
- [Information About MGCP Gateway Support, on page 2](#)
- [How to Configure MGCP Gateway Support, on page 6](#)
- [Configuration Examples for MGCP Gateway Support, on page 10](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring MGCP Gateway Support

The current Media Gateway Control Protocol (MGCP) implementation does not allow the assignment of particular IP addresses for sourcing MGCP commands and media packets, which can cause firewall and security problems. This feature allows you to configure interfaces on which control and media packets can be exchanged. This new functionality allows you to separate signaling from voice by binding control (MGCP signaling) and media (Real-Time Transport Protocol, or RTP voice, fax, and modem) to specific gateway interfaces.

This feature includes new commands that can be used to configure the required interface for MGCP control and control of the required media packets.

Information About MGCP Gateway Support

If the media gateway (MG) uses an IP address, which is the outgoing interface of the MG, the media gateway controller (MGC) identifies the MG entity with that address. If that interface fails, MG sources MGCP from another interface, which is not known to the MGC. Some form of name lookup (host or Domain Name System, or DNS) needs to occur on the MGC at this time. Using the **mgcp bind** command, a loopback interface IP address can be used for sourcing MGCP packets, which is transparent to any interface failure.

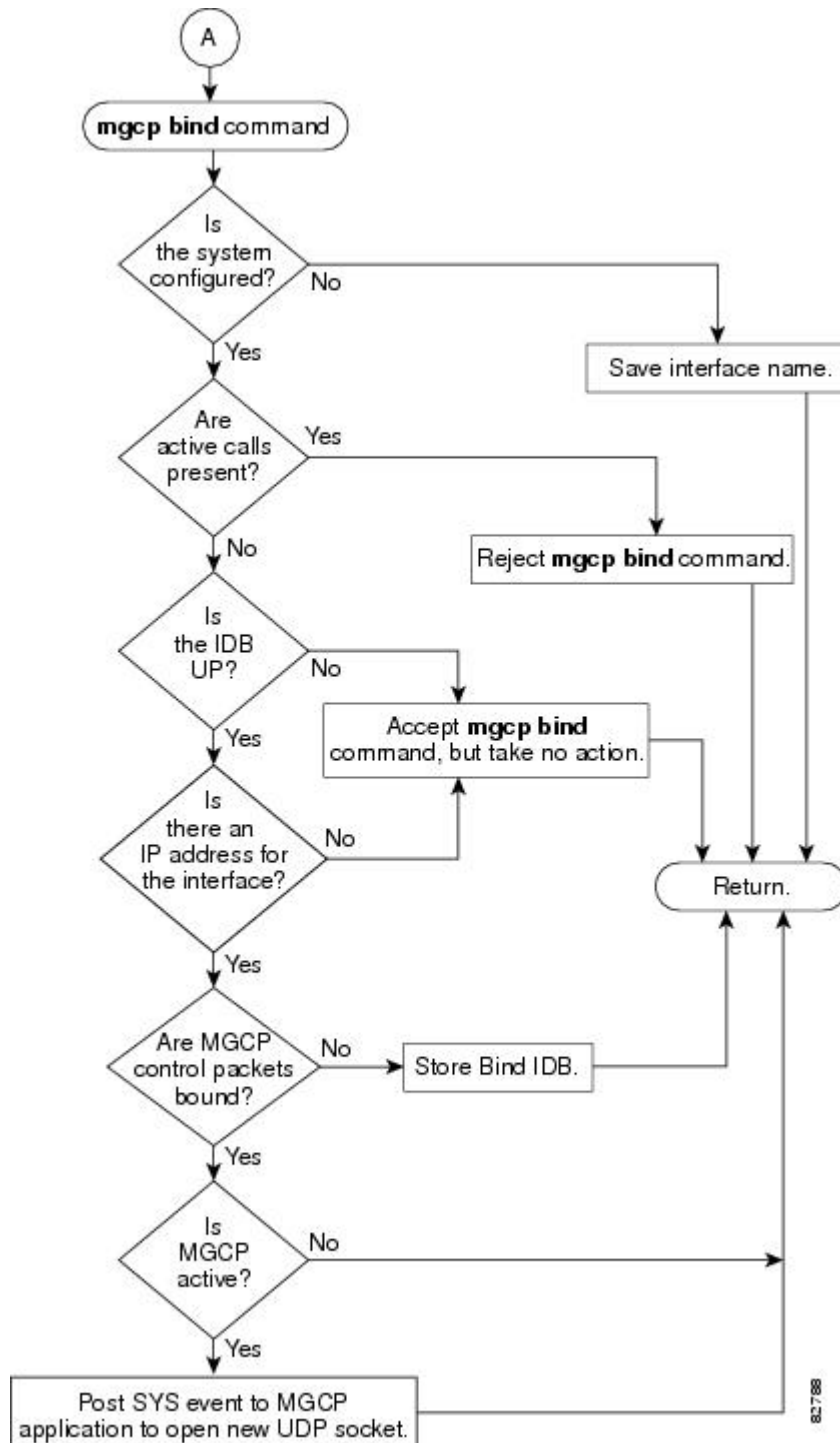
Present implementation of MGCP media uses the "loopback 0" or best available IP address in the order indicated for media. A fixed default loopback 0 address for media streams breaks the dial address pool convention used for most configurations, where dial IP addresses are assigned from the loopback 0 address range. With this feature, it is possible to assign a media bind interface other than loopback 0, which helps dial calls conserve IP addresses.

If you configure the loopback interface for control, you can use routing capability more efficiently. Using the **mgcp bind** command to configure the loopback interface helps in creating redundant MGCP control or media interface.

In the current implementation of MGCP, the source address of MGCP and media control is given by the IP layer. Because of this inconsistency, it is not possible to include a reliable access list or firewall configuration. Using the **mgcp bind** command for both MGCP and media control, you can get a predefined interface or IP address that can be used for security configuration.

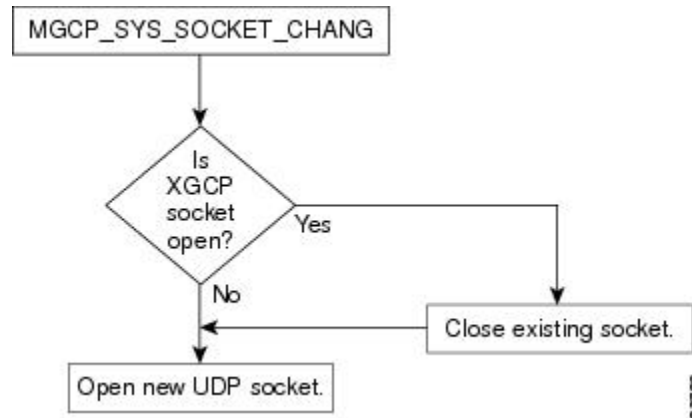
The figure below shows a typical configuration flow using the **mgcp bind** command.

Figure 1: Bind Configuration Flowchart



The figure below shows how the **mgcp bind** command takes effect for MGCP control. When the **mgcp bind** command is configured for MGCP control, the MGCP_SYS_SOCKET_CHANG system event is posted to MGCPAPP. This event is processed by opening a new socket based on the configured interface.

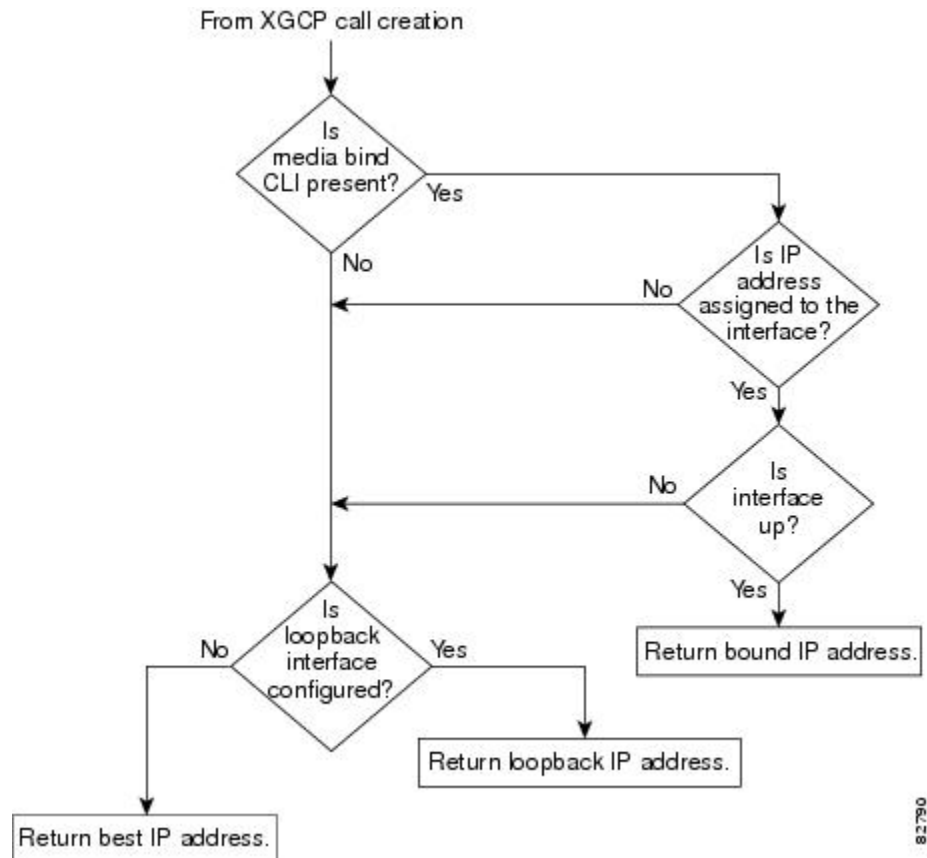
Figure 2: Bind Configuration for Control Flowchart



The time frame for execution of the **mgcp bind** command for media is different from that for control. The figure below shows how the **mgcp bind** command is used for media. In this case, the IP address used for media Session Description Protocol (SDP) negotiation is taken from the configured interface. This flow is not active until an MGCP call is created.

The function call to get an IP address for the media returns a configured interface IP address, a loopback interface IP address, or a best available IP address in the order specified in the figure.

Figure 3: Bind Configuration for Media Flowchart



How to Configure MGCP Gateway Support


Note

If more than 72 end points are configured with MGCP in a single voice gateway, we recommend you to increase the hold-queue size on the interface of the gateway to 300.

Configuring the MGCP Application

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mgcp call-agent** {*dns-name* | *ip-address*} [*port*] [**service-type** *type*] [**version** *protocol-version*]
4. **mgcp**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	mgcp call-agent { <i>dns-name</i> <i>ip-address</i> } [<i>port</i>] [service-type <i>type</i>] [version <i>protocol-version</i>] Example: <pre>Router(config)# mgcp call-agent 209.165.200.225 service-type mgcp version 1.0</pre>	Configures the MGCP protocol and corresponding call agent.
Step 4	mgcp Example: <pre>Router(config)# mgcp</pre>	Enables MGCP on the gateway.
Step 5	exit Example: <pre>Router(config)# exit</pre>	Exits the current mode.

Configuring the bind Command

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mgcp bind {control | media} source-interface *interface-id***
4. **mgcp**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: <pre>Router# configure terminal</pre>	Enters global configuration mode.
Step 3	mgcp bind {control media} source-interface <i>interface-id</i> Example: <pre>Router(config)# mgcp bind {control} source-interface FastEthernet</pre>	Sets a source interface for signaling and media packets.
Step 4	mgcp Example: <pre>Router(config)# mgcp</pre>	Enables MGCP on the gateway.
Step 5	exit Example: <pre>Router(config)# exit</pre>	Exits the current mode.

Troubleshooting Tips

To troubleshoot the MGCP Gateway Support for the Bind Command feature, use the **debug mgcp** command to enable debug traces for MGCP errors, events, media, packets, parser, and call admission control (CAC).

The following example illustrates the output for the **debug mgcp** command with the all keyword:

```
Router# debug mgcp all
20:54:13: MGC stat - 192.168.10.10, total=37, succ=28, failed=8
20:54:13: MGCP Packet received -
```

```

CRCX 55560 s0/ds1-0/1 SGCP 1.1
C: 78980
M: sendrecv
L: a:G.726-16
20:54:13: -- mgcp_parse_packet() - call mgcp_parse_header
- mgcp_parse_header()- Request Verb FOUND CRCX
- mgcp_parse_packet() - out mgcp_parse_header
- SUCCESS: mgcp_parse_packet()-MGCP Header parsing was OK
- mgcp_parse_parameter_lines(), code_str:: 78980, code_len:2, str:1640150312
- mgcp_parse_parameter_lines(str:C: 78980) -num_toks: 19
- mgcp_parse_parameter_lines() check NULL str(78980), in_ptr(C: 78980)
- mgcp_parse_parameter_lines() return Parse function in
mgcp_parm_rules_array[1]
- mgcp_parse_call_id(in_ptr: 78980)
- SUCCESS: mgcp_parse_call_id()-Call ID string(78980) parsing is OK
- mgcp_parse_parameter_lines(), code_str:: sendrecv, code_len:2, str:1640150312
- mgcp_parse_parameter_lines(str:M: sendrecv) -num_toks: 19
- mgcp_parse_parameter_lines() check NULL str(sendrecv), in_ptr(M: sendrecv)
- mgcp_parse_parameter_lines() return Parse function in
mgcp_parm_rules_array[6]
- mgcp_parse_conn_mode(in_ptr: sendrecv)
- mgcp_parse_conn_mode()- tmp_ptr:(sendrecv)
- mgcp_parse_conn_mode(match sendrecv sendrecv)
- mgcp_parse_conn_mode(case MODE_SENDRXCV)
- SUCCESS: Connection Mode parsing is OK
- mgcp_parse_parameter_lines(), code_str:: a:G.726-16, code_len:2,
str:1640150312
- mgcp_parse_parameter_lines(str:L: a:G.726-16) -num_toks: 19
- mgcp_parse_parameter_lines() check NULL str(a:G.726-16), in_ptr(L:
a:G.726-16)
- mgcp_parse_parameter_lines() return Parse function in mgcp_parm_rules_array[5]
- mgcp_parse_con_opts()
- mgcp_parse_codecs()
- SUCCESS: CODEC strings parsing is OK- SUCCESS: Local Connection option
parsing is OK- mgcp_val_mandatory_parms()
20:54:13: - SUCCESS: mgcp_parse_packet()- END of Parsing
20:54:13: MGCP msg 1
20:54:13: mgcp_search_call_by_endpt: endpt = s0/ds1-0/1, new_call = 1
20:54:13: slot=0,ds1=0,ds0=1
20:54:13: search endpoint - New call=1, callp 61C28130
20:54:13: callp: 61C28130, vdbptr: 0, state: 0
20:54:13: mgcp_remove_old_ack:
20:54:13: mgcp_idle_crcx: get capability
passthru is 3
20:54:13: process_request_ev- callp 61C28130, voice_if 61C281A4
20:54:13: process_detect_ev- callp 61C28130, voice_if 61C281A4
process_signal_ev- callp 61C28130, voice_ifp 61C281A4
20:54:13: mgcp_process_quarantine_mode- callp 61C28130, voice_if 61C281A4
20:54:13: mgcp_process_quarantine_mode- new q mode: process=0, loop=0
20:54:13: mgcp_xlat_ccapi_error_code - ack_code_tab_index = 0,
20:54:13: No SDP connection info
20:54:13: mgcp_select_codec - LC option, num codec=1, 1st codec=5
20:54:13: mgcp_select_codec - num supprt codec=11
20:54:13: mgcp_select_codec - LC codec list only
20:54:13: codec index=0, bw=16000, codec=5
20:54:13: selected codec=5mgcp_get_pkt_period: voip_codec=2, pkt_period=0, call
adjust_packetization_period
mgcp_get_pkt_period: voip_codec=2, pkt_period=10, after calling
adjust_packetization_period
20:54:13: selected codec 5
20:54:13: IP Precedence=60
20:54:13: MGCP msg qos value=0mgcp_get_pkt_period: voip_codec=2, pkt_period=0,
call adjust_packetization_period
mgcp_get_pkt_period: voip_codec=2, pkt_period=10, after calling

```



```

adjust_packetization_period
mgcp_new_codec_bytes: voip_codec=2, pkt_period=10, codec_bytes=20
20:54:13: callp : 61C28AE8, state : 2, call ID : 40, event : 5, minor evt:
1640137008
20:54:13: MGCPAPP state machine: state = 2, event = 5
20:54:13: mgcp_call_connect: call_id=40, ack will be sent later.
20:54:13: callp : 61C28AE8, new state : 3, call ID : 40
20:54:14: xlate_ccapi_ev - Protocol is SGCP, change pkg=2
20:54:14: MGCP Session Appl: ignore CCAPI event 22, callp 61C28130
20:54:14: xlate_ccapi_ev - Protocol is SGCP, change pkg=2
20:54:14: callp : 61C28130, state : 2, call ID : 39, event : 5, minor evt: 20
20:54:14: MGCPAPP state machine: state = 2, event = 5
20:54:14: callp : 61C28130, new state : 3, call ID : 39
20:54:14: xlate_ccapi_ev - Protocol is SGCP, change pkg=2
20:54:14: callp : 61C28130, state : 3, call ID : 39, event : 6, minor evt: 20
20:54:14: MGCPAPP state machine: state = 3, event = 6
20:54:14: call_id=39, mgcp_ignore_ccapi_ev: ignore 6 for state 3
20:54:14: callp : 61C28130, new state : 3, call ID : 39
20:54:14: MGCP voice mode event
20:54:14: xlate_ccapi_ev - Protocol is SGCP, change pkg=2
20:54:14: callp : 61C28130, state : 3, call ID : 39, event : 17, minor evt: 0
20:54:14: MGCPAPP state machine: state = 3, event = 17
20:54:14: mgcp_voice_mode_done(): callp 61C28130, major ev 17,
minor ev 0mgcp_start_ld_timer: timer already initialized
20:54:14: send_mgcp_create_ack
20:54:14: map_mgcp_error_code_to_string error_tab_index = 0, protocol version:
2
20:54:14: MGC stat - 1.13.89.3, total=37, succ=29, failed=8
20:54:14: Codec Cnt, 1, first codec 5
20:54:14: First Audio codec, 5, local encoding, 96
20:54:14: -- mgcp_build_packet()-
20:54:14: - mgcp_estimate_msg_buf_length() - 87 bytes needed for header
- mgcp_estimate_msg_buf_length() - 125 bytes needed after checking parameter lines
- mgcp_estimate_msg_buf_length() - 505 bytes needed after cheking SDP lines
20:54:14: --- mgcp_build_parameter_lines() ---
- mgcp_build_conn_id()
- SUCCESS: Conn ID string building is OK
- SUCCESS: Building MGCP Parameter lines is OK
- SUCCESS: building sdp owner id (o=) line
- SUCCESS: building sdp session name (s=) line
- SUCCESS: MGCP message building OK
- SUCCESS: END of building
updating lport with 2427
20:54:14: send_mgcp_msg, MGCP Packet sent --->
200 55560
I: 10
v=0
o=- 78980 0 IN IP4 192.168.10.9
s=Cisco SDP 0
c=IN IP4 192.168.10.9
t=0 0
m=audio 16444 RTP/AVP 96
a=rtpmap:96 G.726-16/8000/1
<---
20:54:14: enqueue_ack: voice_if=61C281A4, ackqhead=0, ackqtail=0,
ackp=61D753E8, msg=61D00010
20:54:14:
mgcp_process_quarantine_after_ack:ack_code=200mgcp_delete_qb_evt_q:cleanup QB
evt q
20:54:14: callp : 61C28130, new state : 4, call ID : 39

```

Verifying MGCP Gateway Support

SUMMARY STEPS

1. Router# **show mgcp**
2. Router# **show ip socket**
3. Router# **show running-configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router# show mgcp	Checks your configuration.
Step 2	Router# show ip socket	Displays IP socket information.
Step 3	Router# show running-configuration	Verifies bind functionality.

Configuration Examples for MGCP Gateway Support

The following is partial output from the **show running-configuration** command indicating that bind is functional on receiving router 172.18.192.204. Updated output for MGCP binding is highlighted under the voice service VoIP indicator.

```
ip subnet-zero
ip ftp source-interface Ethernet0
!
voice service voip
mgcp bind control source-interface FastEthernet0
mgcp bind media source-interface FastEthernet0
!
interface FastEthernet0
ip address 172.18.192.204 255.255.255.0
duplex auto
speed auto
fair-queue 64 256 1000
ip rsvp bandwidth 75000 100
!
```