

SIP Call Hold

The Session Initiation Protocol (SIP) call hold feature in the Session Border Controller (SBC) provides a standard telephony service of putting a caller on hold. If a party in a call wants to put the other party on hold, a party re-invites the other by sending an INVITE request with a modified Session Description Protocol (SDP). When a SIP endpoint wishes to place a call on hold or respond to a call hold re-INVITE, it chooses an appropriate method. SBC modifies call hold SDPs to use any available methods in order to maximize inter-operating with SIP devices.



Note

For ACE SBC Release 3.0.00, this feature is supported in the unified model only.

Feature History for SIP Call Hold

Release	Modification
ACE SBC Release 3.1.00	Support added for SDP call hold interworking.
ACE SBC Release 3.0.00	This feature was introduced on the Cisco 7600 series router along with support for the SBC unified model.

Contents

This module contains the following sections:

- [Information About SIP Call Hold, page 24-1](#)
- [SDP Call Hold Interworking, page 24-3](#)
- [Configuration Examples, page 24-6](#)

Information About SIP Call Hold

SBC accepts a SIP re-INVITE with an SDP, signaling that the sender wishes to put the call on hold. SBC modifies the SDP offer as needed and replaces remote endpoint addresses with known data border element (DBE) media addresses. SBC then forwards the SIP message, containing the modified SDP to the remote endpoint.

If the re-INVITE is rejected by the endpoint going on hold, then the error response is returned to the holding endpoint (the endpoint that initiated the call hold). The media gate on the DBE continues to be connected and media continues to flow as before.

Configuring SIP Call Hold

This section contains the steps for configuring the “no media” timeout duration for on-hold calls.

SUMMARY STEPS

1. **configure**
2. **sbc** *service-name*
3. **sbe**
4. **hold-media-timeout** *timeout*
5. **exit**
6. **show services sbc** *service-name* **sbe hold-media-timeout**
7. **show services sbc** *service-name* **sbe calls**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: host1/Admin# configure	Enables global configuration mode.
Step 2	sbc <i>service-name</i> Example: host1/Admin(config)# sbc mysbc	Enters the mode of an SBC service. Use the <i>service-name</i> argument to define the name of the service.
Step 3	sbe Example: host1/Admin(config-sbc)# sbe	Enters the mode of the signaling border element (SBE) function of the SBC.

	Command or Action	Purpose
Step 4	<p>hold-media-timeout <i>timeout</i></p> <p>Example: host1/Admin(config-sbc-sbe)# hold-media-timeout 7200</p>	<p>The time an SBE will wait after receiving a media timeout notification from the DBE for an on hold call before tearing that call down.</p> <ul style="list-style-type: none"> • When the DBE detects that media has stopped on a call, it will start a timer for the specified duration, using the DBE media timeout command line interface (CLI) command. • If no media flows before this timer expires, then the DBE will send a pin-hole timeout event notification to the SBE. • If the call is on hold, the SBE will set a timer with a duration matching the configured value using this command. • If the call is not taken off hold before the SBE timer expires, then the call will be torn down. <p>The default value for this command is off. Unless a specific duration is set, on hold calls never time out.</p>
Step 5	<p>exit</p> <p>Example: host1/Admin(config-sbc-sbe)# exit</p>	<p>Exits the configuration session and returns to the SBC mode.</p>
Step 6	<p>show services sbc <i>sbc-name</i> sbe hold-media-timeout</p> <p>Example: host1/Admin(config-sbc)# show services sbc mysbc sbe hold-media-timeout</p>	<p>Shows the currently configured duration of the media timeout timer for on-hold calls.</p>
Step 7	<p>show services sbc <i>sbc-name</i> sbe calls</p> <p>Example: host1/Admin(config-sbc)# show services sbc mysbc sbe calls</p>	<p>Lists all the calls on the SBE.</p>

SDP Call Hold Interworking

ACE SBC Release 3.1.00 adds support for SDP call hold interworking. With SDP call hold interworking, there are two ways of setting up call hold using SIP. Either the caller or callee can renegotiate the call characteristics using SDP so that either:

- The connection line is set to the null address, c=IN IP4 0.0.0.0.
- Or the direction attribute for their endpoint so that it does not receive media from the endpoint.
 - If this was previously set to a=sendrecv, the endpoint putting the call on hold sets it to a=sendonly.
 - If this was previously set to a=recvonly, the endpoint putting the call on hold sets it to a=inactive.

Some SIP endpoints support setting the connection line to the null address, some support setting the direction, and some support both approaches. Additionally, some endpoints only respect setting the direction attribute to sendonly or inactive.

With SDP call hold interworking, the SBC supports interoperating with SIP endpoints that support a subset of the above approaches. When the SBC detects that a call is being put on hold in the SDP, it removes any preexisting `c=IN IP4 0.0.0.0` or `a=direction` lines and replaces them with appropriate settings for the endpoint.

If the endpoint putting the call on hold was `sendrecv` or `sendonly`, then the default behavior is to send

- `c=IN IP4 0.0.0.0`
- `a=sendonly`

If the endpoint putting the call on hold was `recvonly` or `inactive`, then the default behavior is to send

- `C=IN IP4 0.0.0.0`
- `a=inactive`

Restrictions for SDP Call Hold Interworking

Review the following restrictions for SDP Call Hold Interworking:

- Music on Hold (MoH) is not supported.
- For middle call service when a call is answered after SBC card switch over (for example, a call hold/resume), the SBC will change the port in SDP. If the peer node does not support SDP port change for middle call service, there will be a middle call service failure.

Configuring SDP Call Hold Interworking

SUMMARY STEPS

1. **configure**
2. **sbc** *service-name*
3. **sbe**
4. **cac-policy-set** *policy-set-id*
5. **first-cac-scope** *scope-name*
6. **first-cac-table** *table-name*
7. **cac-table** *table-name*
8. **match-type** *table-type*
9. **entry** *entry-id*
10. **match-value** *key*
11. **caller-hold-setting** { **hold-c0** | **hold-c0-inactive** | **hold-c0-sendonly** | **hold-sendonly** | **standard** }
12. **action** [**cac-complete** | **next-table** *goto-table-name*]
13. **exit**
14. **exit**
15. **complete**

16. active-cac-policy-set *policy-set-id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: host1/Admin# configure	Enables global configuration mode.
Step 2	sbc <i>service-name</i> Example: host1/Admin(config)# sbc mysbc	Enters the mode of an SBC service. Use the <i>service-name</i> argument to define the name of the service.
Step 3	sbe Example: host1/Admin(config-sbc)# sbe	Enters the mode of the signaling border element (SBE) function of the SBC.
Step 4	cac-policy-set <i>policy-set-id</i> Example: host1/Admin(config-sbc-sbe)# cac-policy-set 1	Enters the submode of CAC policy set configuration within an SBE entity.
Step 5	first-cac-scope <i>scope-name</i> Example: host1/Admin(config-sbc-sbe-cacpolicy)# first-cac-scope global	Configures the scope at which to begin defining limits when performing the admission control stage of policy.
Step 6	first-cac-table <i>table-name</i> Example: host1/Admin(config-sbc-sbe-cacpolicy)# first-cac-table RootCacTable	Configures the name of the first policy table to process when performing the admission control stage of policy.
Step 7	cac-table <i>table-name</i> Example: host1/Admin(config-sbc-sbe-cacpolicy)# cac-table RootCacTable	Creates or configures an admission control table.
Step 8	match-type <i>table-type</i> Example: host1/Admin(config-sbc-sbe-cacpolicy-cactable)# match-type src-account	Configures a new CAC table type that enables the priority of the call to be used as a criterion in CAC policy.
Step 9	entry <i>entry-id</i> Example: host1/Admin(config-sbc-sbe-cacpolicy-cactable)# entry 1	Creates or modifies an entry in a table.

	Command or Action	Purpose
Step 10	match-value <i>key</i> Example: host1/Admin(config-sbc-sbe-cacpolicy-cactable-entry)# match-value fairchild	Configures the match-value of an entry in an admission control table.
Step 11	caller-hold-setting { hold-c0 hold-c0-inactive hold-c0-sendonly hold-sendonly standard } Example: host1/Admin(config-sbc-sbe-cacpolicy-cactable-entry)# caller-hold-setting hold-sendonly	Configures the caller hold settings that are supported.
Step 12	action [cac-complete next-table <i>goto-table-name</i>] Example: host1/Admin(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete	Specifies the action to take if this routing entry is chosen.
Step 13	exit Example: host1/Admin(config-sbc-sbe-cacpolicy-cactable-entry)# exit	Exits the configuration session and returns to the SBC mode.
Step 14	exit Example: host1/Admin(config-sbc-sbe-cacpolicy-cactable)# exit	Exits the configuration session and returns to the SBC mode.
Step 15	complete Example: host1/Admin(config-sbc-sbe-cacpolicy)# complete	Completes the CAC-policy or call-policy set after committing the full set.
Step 16	active-cac-policy-set <i>policy-set-id</i> Example: host1/Admin (config-sbc-sbe)# active-cac-policy-set 1	Sets the active CAC-policy-set within an SBE entity.

Configuration Examples

The section contains configuration examples.

Example of Configuring SIP Call Hold

The following command configures the SBE to wait for two hours after receiving the last media packet on an on-hold call before cleaning up the call resources.

```
host1/Admin# configure
host1/Admin(config)# sbc mysbc
host1/Admin(config-sbc)# sbe
host1/Admin(config-sbc-sbe)# hold-media-timeout 7200
```

Example of Configuring SDP Call Hold Interworking

In the example below, Fairchild Foods has replaced all the phones in their offices. The new phones support setting `a=sendonly` and `c=IN IP4 0.0.0.0` to place a call on hold; they do not support setting `a=inactive`. You now want to reconfigure the SBC to work with these phones without changing the behavior for other customers. This change creates new policies at the account scope for all events, so that calls in which Fairchild Foods phones are involved are put on hold appropriately.

The following configuration changes will make sure that Fairchild phone doesn't receive `a=inactive` in SDP when Fairchild is the source account and the callee puts the call on hold:

```
host1/Admin# configure
host1/Admin(config)# sbc mysbc
host1/Admin(config-sbc)# sbe
host1/Admin(config-sbc-sbe)# cac-policy-set 1
host1/Admin(config-sbc-sbe-cacpolicy)# first-cac-scope global
host1/Admin(config-sbc-sbe-cacpolicy)# first-cac-table callhold-src-settings
host1/Admin(config-sbc-sbe-cacpolicy)# cac-table callhold-src-settings
host1/Admin(config-sbc-sbe-cacpolicy-cactable)# match-type src-account
host1/Admin(config-sbc-sbe-cacpolicy-cactable)# entry 1
host1/Admin(config-sbc-sbe-cacpolicy-cactable-entry)# match-value fairchild
host1/Admin(config-sbc-sbe-cacpolicy-cactable-entry)# caller-hold-setting hold-c0-sendonly
host1/Admin(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
host1/Admin(config-sbc-sbe-cacpolicy-cactable-entry)# exit
host1/Admin(config-sbc-sbe-cacpolicy-cactable)# exit
host1/Admin(config-sbc-sbe-cacpolicy)# complete
```

