



SIP Call Hold

The Session Initiation Protocol (SIP) call hold feature in Cisco Unified Border Element (SP Edition) 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. Cisco Unified Border Element (SP Edition) modifies call hold SDPs to use any available methods in order to maximize inter-operating with SIP devices.



Note

For Cisco IOS XE Release 2.4, this feature is supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model* at:

http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu_book.html.

For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or a Cisco IOS master commands list.

Feature History for SIP Call Hold

Release	Modification
Cisco IOS XE Release 2.4	This feature was introduced on the Cisco IOS XR along with support for the unified model.

Contents

This module contains the following sections:

- [Information About SIP Call Hold in Cisco Unified Border Element \(SP Edition\)](#), page 46-2
- [How to Configure SIP Call Hold](#), page 46-2
- [SDP Call Hold Interworking](#), page 46-3
- [Configuration Examples](#), page 46-10

Information About SIP Call Hold in Cisco Unified Border Element (SP Edition)

Cisco Unified Border Element (SP Edition) accepts a SIP re-INVITE with an SDP, signaling that the sender wishes to put the call on hold. Cisco Unified Border Element (SP Edition) modifies the SDP offer as needed and replaces remote endpoint addresses with known data border element (DBE) media addresses. Cisco Unified Border Element (SP Edition) 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.

How to Configure SIP Call Hold

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

Configuring SIP Call Hold

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: Router# configure	Enables global configuration mode.
Step 2	sbc <i>service-name</i> Example: Router(config)# sbc mysbc	Enters the mode of an SBC service. Use the <i>service-name</i> argument to define the name of the service.

	Command or Action	Purpose
Step 3	sbe Example: Router(config-sbc)# sbe	Enters the mode of the signaling border element (SBE) function of the SBC.
Step 4	hold-media-timeout <i>timeout</i> Example: Router(config-sbc-sbe)# hold-media-timeout 7200	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. - 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. The default value for this command is off. Unless a specific duration is set, on hold calls never time out.
Step 5	end Example: Router(config-sbc-sbe)# end	Exits the configuration session and enters Privileged EXEC mode.
Step 6	show sbc <i>sbc-name</i> sbe hold-media-timeout Example: Router# show sbc mysbc sbe hold-media-timeout	Shows the currently configured duration of the media timeout timer for on-hold calls.
Step 7	show sbc <i>sbc-name</i> sbe calls Example: Router# show sbc mysbc sbe calls	Lists all the calls on the SBE.

SDP Call Hold Interworking

Cisco IOS XE Release 2.4 introduces 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 to 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, Cisco Unified Border Element (SP Edition) supports interoperating with SIP endpoints that support a subset of the above approaches. When Cisco Unified Border Element (SP Edition) 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`



Note

Music on hold is supported using SDP call hold interworking.

Configuring SDP Call Hold Interworking

SUMMARY STEPS

1. **configure terminal**
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. **table-type {*policy-set* | *limit* {*list of limit tables*}}**
9. **entry *entry-id***
10. **cac-scope {*list of scope options*}**
11. **match-value *key***
12. **caller-hold-setting {*hold-c0* | *hold-c0-inactive* | *hold-c0-sendonly* | *hold-sendonly* | *standard*}**
13. **action [*cac-complete* | *next-table goto-table-name*]**
14. **exit**
15. **exit**
16. **complete**
17. **active-cac-policy-set *policy-set-id***

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: Router# configure terminal	Enables global configuration mode.
Step 2	sbc service-name Example: Router(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: Router(config-sbc)# sbe	Enters the mode of the signaling border element (SBE) function of the SBC.
Step 4	cac-policy-set policy-set-id Example: Router(config-sbc-sbe)# cac-policy-set 1	Enters the submode of CAC policy set configuration within an SBE entity.
Step 5	first-cac-scope scope-name Example: Router(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. The <i>scope-name</i> argument configures the scope at which limits should be initially defined. Possible values are: • adj-group • call • category • dst-account • dst-adj-group • dst-adjacency • dst-number • global • src-account • src-adj-group • src-adjacency • src-number Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies.

	Command or Action	Purpose
Step 6	first-cac-table <i>table-name</i> Example: Router(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: Router(config-sbc-sbe-cacpolicy)# cac-table RootCacTable	Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.

	Command or Action	Purpose
Step 8	table-type {policy-set limit {<i>list of limit tables</i>}} Example: Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit event-type	Configures the table type of a CAC table within the context of an SBE policy set. <i>list of limit tables</i> can be one of the following values: • account—Compare the name of the account. • adj-group—Compare the name of the adjacency group. • adjacency—Compare the name of the adjacency. • all—No comparison type. All events match this type. • call-priority—Compare with call priority. • category—Compare the number analysis assigned category. • dst-account—Compare the name of the destination account. • dst-adj-group—Compare the name of the destination adjacency group. • dst-adjacency—Compare the name of the destination adjacency. • dst-prefix—Compare the beginning of the dialed digit string. • event-type—Compare with CAC policy event types. • src-account—Compare the name of the source account. • src-adj-group—Compare the name of the source adjacency group. • src-adjacency—Compare the name of the source adjacency. • src-prefix—Compare the beginning of the calling number string. Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group.
Step 9	entry entry-id Example: Router(config-sbc-sbe-cacpolicy-cactable)# entry 1	Creates or modifies an entry in a table.

Command or Action	Purpose
Step 10 <code>cac-scope {list of scope options}</code> Example: Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope call	Configures the scope within each of the entries at which limits are applied in a policy set table. • <i>list of scope options</i>—Specifies one of the following strings used to match events: – account—Events that are from the same account. – adjacency—Events that are from the same adjacency. – adj-group—Events that are from members of the same adjacency group. – call—Scope limits are per single call. – category—Events that have same category. – dst-account—Events that are sent to the same account. – dst-adj-group—Events that are sent to the same adjacency group. – dst-adjacency—Events that are sent to the same adjacency. – dst-number—Events that have same destination. – global—Scope limits are global – src-account—Events that are from the same account. – src-adj-group—Events that are from the same adjacency group. – src-adjacency—Events that are from the same adjacency. – src-number—Events that have the same source number. – sub-category—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category. – sub-category-pfx—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category prefix. – subscriber—The limits specified in this scope apply to all events sent to or received from individual subscribers (a device that is registered with a Registrar server).

	Command or Action	Purpose
Step 11	match-value <i>key</i> Example: Router(config-sbc-sbe-cacpolicy-cactable-entry) # match-value call-update	Specifies the keyword used to match events. The format of the key is determined by the table-type. If you configure either an event-type or call-priority Limit table, then you only see the keyword options that apply for that type of Limit table. For Limit event-type tables (table-type limit event-type), the match value keyword options are the following: • call-update—Compare the beginning of the calling number string. • endpoint-reg—Compare the name of the destination adjacency. • new-call—Compare the beginning of the dialed digit string. For Limit call-priority tables (table-type limit call-priority), the match value keyword options are the following: • critical—Match calls with resource priority 'critical.' • flash—Match calls with resource priority 'flash'. • flash-override—Match calls with resource priority 'flash-override.' • immediate—Match calls with resource priority 'immediate.' • priority—Match calls with resource priority 'priority.' • routine—Match calls with resource priority 'routine.' For all other Limit tables, enter a name or digit string. WORD—Name or digit string to match. (Max Size 255).
Step 12	caller-hold-setting {hold-c0 hold-c0-inactive hold-c0-sendonly hold-sendonly standard} Example: Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller-hold-setting hold-sendonly	Configures the caller hold settings that are supported.
Step 13	action [cac-complete next-table goto-table-name] Example: Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete	Specifies the action to take if this routing entry is chosen.

	Command or Action	Purpose
Step 14	exit Example: Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit	Exits the cactable entry configuration mode and enters the cactable mode.
Step 15	exit Example: Router(config-sbc-sbe-cacpolicy-cactable)# exit	Exits the cactable configuration mode and enters the cacpolicy mode.
Step 16	complete Example: Router(config-sbc-sbe-cacpolicy)# complete	Completes the CAC-policy or call-policy set after committing the full set.
Step 17	active-cac-policy-set <i>policy-set-id</i> Example: Router (config-sbc-sbe)# active-cac-policy-set 1	Sets the active CAC-policy-set within an SBE entity.

Configuration Examples

This 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.

```
Router# configure
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# hold-media-timeout 7200
```

Example of Configuring SDP Call Hold Interworking

In the example below, Fairchild Foods have 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 Cisco Unified Border Element (SP Edition) 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 Fairchild phone doesn't receive a=inactive in SDP when Fairchild is the source account and the callee puts the call on hold.

```
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
```

```
Router(config-sbc-sbe-cacpolicy)# first-cac-scope global
Router(config-sbc-sbe-cacpolicy)# first-cac-table callhold-src-settings
Router(config-sbc-sbe-cacpolicy)# cac-table callhold-src-settings
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit src-account
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value fairchild
Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-hold-setting hold-c0-sendonly
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# exit
Router(config-sbc-sbe-cacpolicy)# complete
```

The following configuration changes will make sure Fairchild phone doesn't receive a=inactive in SDP when Fairchild is the destination account and the caller puts the call on hold.

```
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-scope global
Router(config-sbc-sbe-cacpolicy)# first-cac-table callhold-dst-settings
Router(config-sbc-sbe-cacpolicy)# cac-table callhold-dst-settings
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit dst-account
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value fairchild
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-hold-setting hold-c0-sendonly
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# exit
Router(config-sbc-sbe-cacpolicy)# complete
```

