CHAPTER 6

Implementing Adjacencies on Cisco Unified Border Element (SP Edition)

Accounts and adjacencies are the key objects used to control signaling. An account represents a service relationship with a remote organization on the signaling border element (SBE), with which Cisco Unified Border Element (SP Edition) will interact. Within each account, the user defines one or more signaling adjacencies, which connect Cisco Unified Border Element (SP Edition) to devices within that organization. The account is used to:

- Define customer-specific admission control
- Define routing policy configurations
- Organize billing records

An adjacency represents a signaling relationship with a remote call agent. There is one adjacency defined per external call agent. The adjacency is used to define protocol-specific parameters as well as admission control and routing policy. Each adjacency belongs within an account.

Each incoming call is matched to an adjacency, and each outgoing call is routed out over a second adjacency. Adjacencies can also be associated with a media gateway location, so that the most appropriate virtual data border element (vDBE) can be selected for a given call leg. Typically, an Cisco Unified Border Element (SP Edition) has at least one account representing the internal network.

You can assign each adjacency to an adjacency group, so you can enable and disable features per interface. For example, you can turn off high bandwidth features on all adjacencies to customers on a known low-bandwidth link.

This chapter also discusses the SIP Over Transport Layer Security (TLS) feature, an encryption feature that provides a secure, encrypted transport to carry all SIP messages from the caller to the callee's domain.

Note 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).

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

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 Implementing Adjacencies and SIP Over TLS

Release	Modification
Cisco IOS XE Release 2.4	This feature and SIP over TLS were introduced on the Cisco IOS XR along with support for the unified model.
Cisco IOS XE Release 2.6	The following features were added: • Configurable Mutual TLS Authentication Per Interface. • TLS Transport Parameter in Record Route Headers.
Cisco IOS XE Release 3.1S	The Redundant Peer Addresses feature was added.
Cisco IOS XE Release 3.2S	The SIP peer availability detection feature was added. The Public Key Infrastructure (PKI) High Availability (HA) support was added.

Contents

This module contains the following sections:

Prerequisites for Implementing Adjacencies

The following prerequisite is required to implement adjacencies:

- Before implementing adjacencies, Cisco Unified Border Element (SP Edition) must already be configured.

Restrictions

H.323 adjacencies are not supported in Cisco IOS XE Release 2.4 and earlier.

Information About Implementing Adjacencies

Adjacencies are used to enable call signaling between the SBE and other voice over IP (VoIP) devices. Cisco Unified Border Element (SP Edition) supports adjacencies in Session Initiation Protocol (SIP) network deployments.

In a SIP network, the devices might be user agents, proxies, softswitches, or back-to-back user agents (B2BUAs). When you configure a SIP adjacency, the SBE functions as a B2BUA within the SIP network.

Adjacencies can represent both trunking and subscriber signaling relationships. The network topology and configuration of an adjacency determine its role.

The adjacency accepts packets from either the UDP or TCP socket specified in the signaling port configuration line. For SIP, the default is port 5060. When sending packets out the adjacency, the transport used is specified using the **preferred-transport** [**tcp** | **udp**] command. The default is to use UDP. Note that there is no dependency between the input and output adjacencies. It is valid to have one adjacency use TCP for the signaling and the other use UDP.

Further overview details about implementing adjacencies are described in the following sections:

- Properties Common to SIP Adjacencies
- About SIP Adjacencies in the Deployment
- How Adjacencies Affect Media Routing

Properties Common to SIP Adjacencies

The following properties are common to SIP adjacencies:

- Adjacencies are known by name. The name makes it easy for a Cisco Unified Border Element (SP Edition) policy to reference the adjacency.
- An adjacency has a local address and port for incoming call setup.
- An adjacency has a peer address and port. This is the point of contact for outgoing calls. In the SIP case, this is only true if the "force-signaling-peer" option is set for that adjacency.
- An adjacency forms the output of a routing policy decision. In other words, the routing phase for a call results in selection of an outgoing adjacency for that call. Normally, adjacency selection is done based on a destination telephone number prefix. However, two adjacencies can also be bridged together by using a source adjacency as a routing input.

About SIP Adjacencies in the Deployment

Figure 6-1 shows a simple SIP network where:

- SIP subscribers register with the SIP proxy, which acts as a single point of contact for all of them.
- The softswitch is a gateway between the SIP network and the public switched telephone network (PSTN).

- The softswitch routing policy assigns a particular phone prefix to each SIP proxy, allowing calls from the PSTN network to be routed through the proxy to a given subscriber. (In other deployments, subscribers may register directly with a softswitch without going through a proxy first.)

Figure 6-1 SIP Network

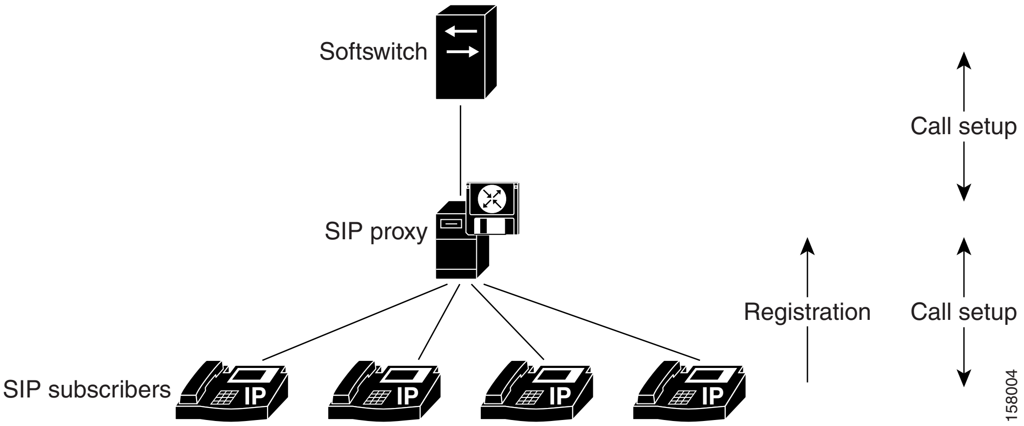


Figure 6-2 shows placement of a Cisco Unified Border Element (SP Edition) in two possible positions within the SIP network, with the adjacencies noted. Each adjacency enables call setup to one or more neighboring devices, as follows:

- ADJ_SIP1A allows call setup between SBC1 and the softswitch.
- ADJ_SIP1B allows call setup between SBC1 and the proxy.
- ADJ_SIP2A allows call setup between SBC2 and the proxy.
- ADJ_SIP_SUBSCRIBERS allows call setup between SBC2 and the subscribers.

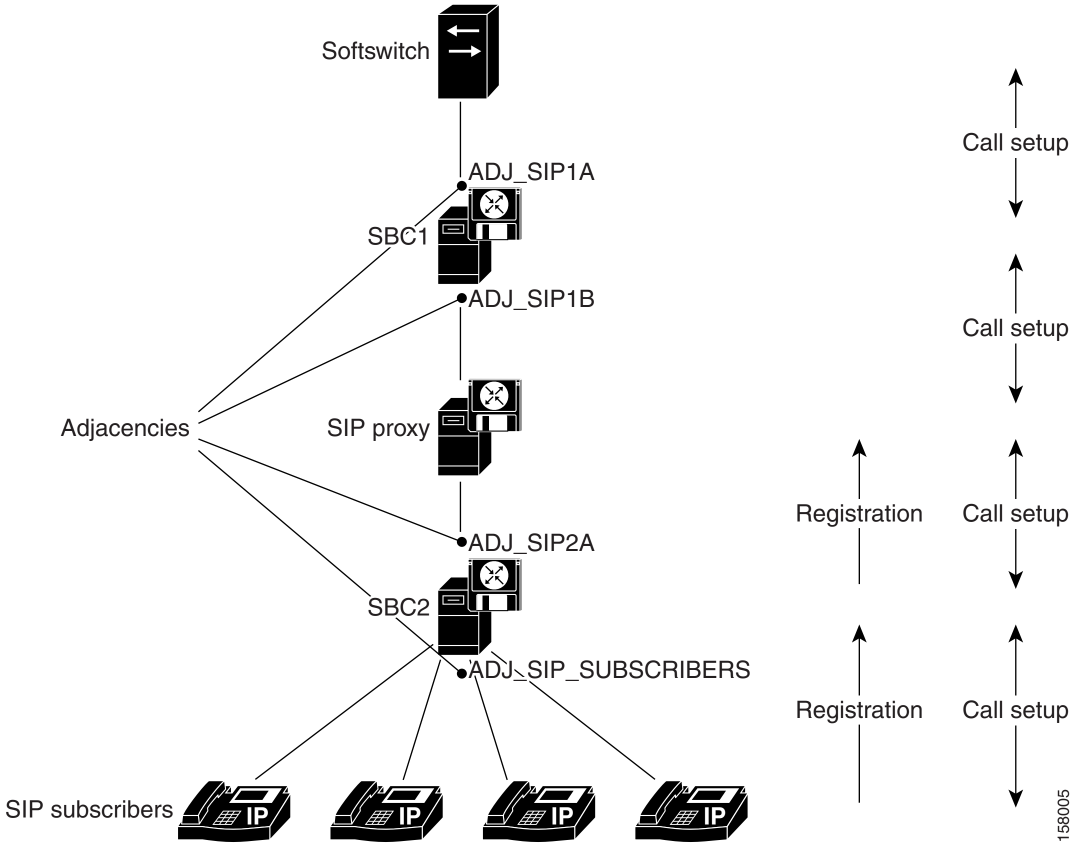
In the case of SBC2, SIP registrations are being routed through the SBC. Registrations received on ADJ_SIP_SUBSCRIBERS are being routed to the proxy over ADJ_SIP2A.

The key difference between subscriber and nonsubscriber adjacencies is that:

- Nonsubscriber adjacencies have a configured single point of contact, the peer address for the adjacency.
- Subscriber adjacencies do not have a single point of contact and are instead configured to accept registrations.

SIP registrations require a routing policy to determine which is the correct outgoing adjacency for a given registration. This works in a very similar way to a call routing policy. See the procedures described in the Implementing Cisco Unified Border Element (SP Edition) Policies module.

Figure 6-2 Adjacencies in a SIP Network Deployment



How Adjacencies Affect Media Routing

For a distributed Cisco Unified Border Element (SP Edition) deployment, each adjacency is configured with a *media location*. The media location is an ID used to select the data border elements (DBEs) suitable for relaying media traffic for calls set up over the adjacency.

If a call is routed out over the same or different adjacency, the media may bypass a DBE. The media bypass feature allows the media packets to bypass the Cisco Unified Border Element (SP Edition) to enable the endpoints to communicate directly to each other. Media packets flow directly without going through the DBE component of the SBC after the call signaling is done. Signaling packets still flow through the SBC as usual.

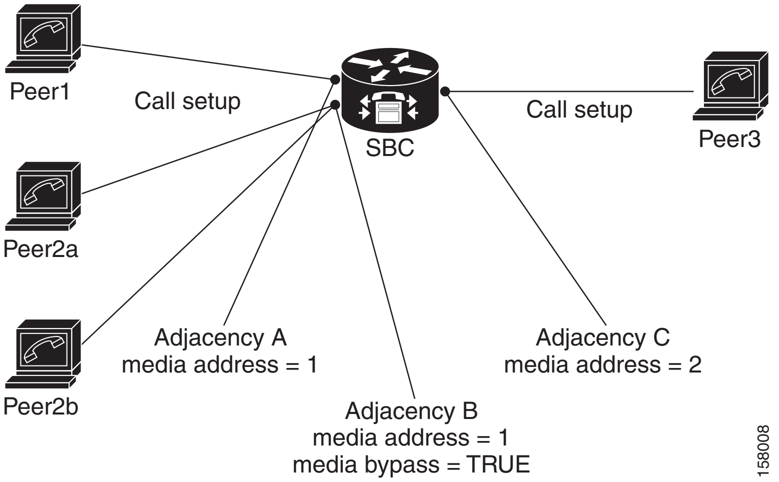
The configuration is set per adjacency, and allows media bypass across different adjacencies. Media-bypass configuration is enabled under adjacency configuration. Media bypass is useful when two endpoints are on the same subnet, but the DBE is located elsewhere on the network.

Figure 6-3 and Figure 6-4 illustrate how adjacency configuration controls media routing. In this example:

- Adjacency A connects to Peer1
- Adjacency B connects to Peer2a and 2b
- Adjacency C connects to Peer3

Adjacencies A and B are configured with media location 1. In other words, calls routed over them will use the same DBE (or set of DBEs) for media. Adjacency C is configured with media location 2.

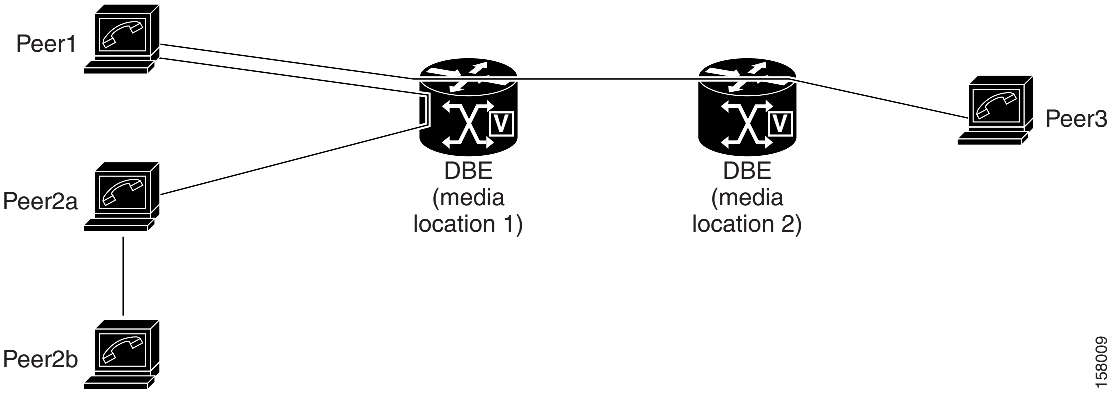
Figure 6-3 How Adjacency Configuration Controls Media Routing



Now consider three calls: Peer1-Peer3, Peer1-Peer2a, and Peer2a-Peer2b. The media for these calls is routed as shown in Figure 6-4.

- The first call traverses two adjacencies with different media locations. Its media is relayed through two DBEs.
- The second call traverses two adjacencies with the same media location. Its media is relayed through a single DBE.
- The third call traverses a single adjacency with media by pass enabled. Its media is sent directly between the two peers without involving a DBE.

Figure 6-4 Media Routing for Three Calls: Peer1-Peer3, Peer1-Peer2a, and Peer2a-Peer2b



How to Implement Adjacencies

Adjacencies are the key objects used to control signaling. The user defines one or more signaling adjacencies, which connect the Cisco Unified Border Element (SP Edition) to devices within that organization. Each incoming call is matched to an adjacency, and each outgoing call is routed out over an adjacency. The adjacencies are then attached to the appropriate account. Adjacencies can be associated with a media gateway DBE location, so that the most appropriate DBE can be selected to route media for a given call leg.

Note The default behavior for Cisco Unified Border Element (SP Edition) is to route INVITE requests to the device specified in the Request URI. If instead, the user wishes requests to be routed to the signaling peer, then 'force-next-hop' behavior should be enabled by configuring the **force-signaling-peer** command on the outbound adjacency.

The following sections describe implementing a SIP adjacency, depending on your implementation requirements:

- Configuring Force-Signaling-Peer Adjacency, page 6-7
- Configuring a SIP Adjacency, page 6-8
- Assigning SIP Adjacencies to Adjacency Groups, page 6-13

Configuring Force-Signaling-Peer Adjacency

This task configures a force-signaling-peer adjacency.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **force-signaling-peer**
6. **attach**
7. **exit**

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 umsbc-node10`
		Enters the mode of an SBC service.
		Use the *service-name* argument to define the name of the service.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	**`adjacency sip`** *`adjacency-name`*
	Example:
	`Router(config-sbc-sbe)# adjacency sip 2651XM-5`
		Enters the mode of an SBE SIP adjacency.
		Use the *adjacency-name* argument to define the name of the SIP adjacency.

Step 5	**`force-signaling-peer`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# force-signaling-peer`
		Forces SIP messages to go to the configured signaling peer.

Step 6	**`attach`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# attach`
		Attaches the adjacency.

Step 7	**`exit`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# exit`
		Exits the **sip** mode to the **sbe** mode.

Configuring a SIP Adjacency

You can only modify adjacencies when the adjacency is detached. Before modifying an adjacency, you can detach the adjacency first with the **no attach** command. The adjacency stays in the going down state when a call is active or when the ping enable feature is running. During this state, existing calls are not torn down and new calls are not accepted. The adjacency does not go to detached state until all calls have ended. An adjacency cannot be attached until the adjacency is in detached state.

If you wish to override the option to wait till active calls on the adjacency end, the adjacency can be detached immediately using the following commands:

- **no attach force abort**—Executes a forced detach, tearing down calls without signaling their end.
- **no attach force normal**—Executes a forced detach, tearing down calls gracefully.

To check the state of the adjacency, you can use the **show sbc sbe adjacencies** command.

Caution Adjacencies can only be modified when the status is detached. Before modifying an adjacency, use the **no attach** command first.

Note For User-to-Network Interface (UNI) registration support for a SIP inherit profile, you have the option of using the default value or a preset-access or a preset-core value. When using the default value for those adjacencies without specific per adjacency configuration, the **sip inherit profile preset-standard-non-ims** command in the SBE configuration mode (config-sbc-sbe) is applied to the adjacencies by default, and UNI registration support is enabled for this default configuration.When configuring a a preset-access or a preset-core value, use the **inherit profile preset-p-cscf-access** command on the adjacency facing subscribers and the **inherit profile preset-p-cscf-core** command on the adjacency facing the SIP proxy. If you use other combinations (for example, if both the inbound and outbound adjacencies are configured as preset-core, Cisco Unified Border Element (SP Edition) will not store the registration information, nor will it rewrite the Contact: header to make sure it's on the signaling path of future messages.

This task configures two session initiation protocol (SIP) adjacencies. The first adjacency is configured for a gateway/endpoint. The second adjacency is configured with proxy/softswitch.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-name*
3. **sbe**
4. **sip inherit profile** {**preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-standard-non-ims**}
5. **adjacency sip** *adjacency-name*
6. **signaling-address ipv4** *ipv4_IP_address*
7. **signaling-port** *port_num*
8. **remote-address ipv4** *ipv4_IP_address/prefix*
9. **signaling-peer** *peer_address*
10. **signaling-peer-port** *port_num*
11. **account** *account-name*
12. **registration rewrite-register**
13. **attach**
14. **exit**
15. **adjacency sip** *adjacency-name*
16. **inherit profile {preset-access | preset-core | preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-peering | preset-standard-non-ims}**
17. **signaling-address ipv4** *ipv4_IP_address*
18. **signaling-port** *port_num*

19. **remote-address ipv4** *ipv4_IP_address/prefix*
20. **fast-register disable**
21. **signaling-peer** *peer_name*
22. **signaling-peer-port** *port_num*
23. **account** *account-name*
24. **registration target address host_address**
25. **registration target port port_num**
26. **attach**
27. **exit**
28. **end**
29. **show**

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** `Router(config)# sbc mysbc`	Enters the mode of an SBC service. Use the *service-name* argument to define the name of the service.
Step 3	**sbe** **Example:** `Router(config-sbc)# sbe`	Enters the mode of an SBE entity within an SBC service.

Step 4	**sip inherit profile** {**preset-ibcf-ext-untrusted** | **preset-ibcf-external** | **preset-ibcf-internal** | **preset-p-cscf-access** | **preset-p-cscf-core** | **preset-standard-non-ims**}
	Example:
	`Router(config-sbc-sbe)# sip inherit profile preset-standard-non-ims`
		Configures the global default inherit profile for all adjacencies.

Step 5	**adjacency sip** *adjacency-name* **Example:** `Router(config-sbc-sbe)# adjacency sip sipGW`	Enters the mode of an SBE SIP adjacency. Use the *adjacency-name* argument to define the name of the service.
Step 6	**signaling-address ipv4** *ipv4_IP_address* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.88.141.3`	Specifies the local IPv4 signaling address of the SIP adjacency.

	Command or Action	Purpose
Step 7	**signaling-port** *port_num* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-port 5060`	Specifies the local signaling port of the SIP adjacency.
Step 8	**remote-address ipv4** *ipv4_IP_address/prefix* **Example:** `Router(config-sbc-sbe-adj-sip)# remote-address ipv4 10.10.121.0/24`	Restricts the set of remote signaling peers contacted over the adjacency to those with the given IP address prefix.
Step 9	**signaling-peer** *peer_address* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-peer 10.10.121.10`	Specifies the remote signaling peer for the SIP adjacency to use.
Step 10	**signaling-peer-port** *port_num* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060`	Specifies the remote signaling-peer port for the SIP adjacency to use.
Step 11	**account** *account_name* **Example:** `Router(config-sbc-sbe-adj-sip)# account iosgw`	Defines the SIP adjacency as belonging to an account on an SBE.
Step 12	**registration rewrite-register** **Example:** `Router(config-sbc-sbe-adj-sip)# registration rewrite-register`	Configures SIP REGISTER request rewriting.
Step 13	**attach** **Example:** `Router(config-sbc-sbe-adj-sip)# attach`	Attaches the adjacency.
Step 14	**exit** **Example:** `Router(config-sbc-sbe-adj-sip)# exit`	Exits **adj-sip** mode to **sbe** mode.
Step 15	**adjacency sip** *adjacency-name* **Example:** `Router(config-sbc-sbe)# adjacency sip sipPROXY`	Enters the mode of an SBE SIP adjacency. Use the *adjacency-name* argument to define the name of the service.

	Command or Action	Purpose

Step 16	**inherit profile** {**preset-access** | **preset-core** | **preset-ibcf-ext-untrusted** | **preset-ibcf-external** | **preset-ibcf-internal** | **preset-p-cscf-access** | **preset-p-cscf-core** | **preset-peering** | **preset-standard-non-ims**}
	Example:
	`Router(config-sbc-sbe-adj-sip)# inherit profile preset-standard-non-ims`
		Configures an inherit profile for the SIP adjacency.

Step 17	**signaling-address ipv4** *ipv4_IP_address* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.88.141.11`	Specifies the local IPv4 signaling address of the SIP adjacency.
Step 18	**signaling-port** *port_num* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-port 5060`	Specifies the local signaling port of the SIP adjacency.
Step 19	**remote-address ipv4** *ipv4_IP_address/prefix* **Example:** `Router(config-sbc-sbe-adj-sip)# remote-address ipv4 200.200.200.0/24`	Restricts the set of remote signaling peers contacted over the adjacency to those with the given IP address prefix.
Step 20	**fast-register disable** **Example:** `Router(config-sbc-sbe-adj-sip)# fast-register disable`	Disables fast register support on the SIP adjacency.
Step 21	**signaling-peer** *peer_address* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-peer 200.200.200.98`	Specifies the remote signaling peer for the SIP adjacency to use.
Step 22	**signaling-peer-port** *port_num* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060`	Specifies the remote signaling-peer port for the SIP adjacency to use.
Step 23	**account** *account_name* **Example:** `Router(config-sbc-sbe-adj-sip)# account COREvlan`	Defines the SIP adjacency as belonging to an account on an SBE.

	Command or Action	Purpose
Step 24	**registration target address** *host_address* **Example:** `Router(config-sbc-sbe-adj-sip)# registration target address 200.200.200.98`	Sets the address to use if rewriting an outbound SIP REGISTER request.
Step 25	**registration target port** *port_num* **Example:** `Router(config-sbc-sbe-adj-sip)# registration target port 5060`	Sets the port to use if rewriting an outbound SIP REGISTER request.
Step 26	**attach** **Example:** `Router(config-sbc-sbe-adj-sip)# attach`	Attaches the adjacency.
Step 27	**exit** **Example:** `Router(config-sbc-sbe-adj-sip)# exit`	Exits **adj-sip** mode to **sbe** mode.

Step 28	**end**
	Example:
	`Router(config-sbc-sbe)# end`
		Exits the sbe mode and returns to Privileged EXEC mode.

Step 29	**show** **Example:** `Router# show`	Shows contents of configuration.

Assigning SIP Adjacencies to Adjacency Groups

Use the procedure in this section to assign an SIP adjacency to an adjacency group.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **group** *adjacency-group-name*
6. **end**
7. **show**

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 *service-name* argument to define the name of the service.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	**`adjacency sip`** *`adjacency-name`*
	Example:
	`Router(config-sbc-sbe)# adjacency sip sipGW`
		Enters the mode of an SBE SIP adjacency.
		Use the *adjacency-name* argument to define the name of the service.

Step 5	**`group`** *`adjacency-group-name`*
	Example:
	`Router(config-sbc-sbe-adj-sip)# group InternetEth0`
		Assigns the SIP adjacency to an adjacency group.
		Use the *adjacency-group-name* argument to define the group name.

Step 6	**`end`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# end`
		Exits **adj-sip** mode to **sbe** mode and returns to Privileged EXEC mode.

Step 7	**`show`**
	Example:
	`Router# show`
		Shows contents of configuration.

Configuration Examples for Implementing Adjacencies

This section provides the following configuration example:

- Configuring a SIP Adjacency: Example, page 6-15

Configuring a SIP Adjacency: Example

The following example configures two SIP adjacencies. The first adjacency is configured for a gateway/endpoint. The second adjacency is configured with proxy/softswitch.

1. Activate SBE, as follows:

```
sbc sip-signal
 sbe
  activate
  exit
```

2. Activate DBE, as follows:

```
sbc mySbc dbe
 media-address ipv4 88.88.141.2
 activate
 exit
```

3. Create the SIP adjacencies, as follows:

```
sbc sip-signal
 sbe
```

4. Create the SIP adjacency for gateway/endpoint:

```
  adjacency sip sipGW
   signaling-address ipv4 88.88.141.3
   signaling-port 5060
   remote-address ipv4 10.10.121.0/24
   signaling-peer 10.10.121.10
   signaling-peer-port 5060
   account iosgw
   registration rewrite-register
  attach
  exit
  !
  !
```

5. Create the SIP adjacency for proxy/softswitch:

```
  adjacency sip sipPROXY
   signaling-address ipv4 88.88.141.11
   signaling-port 5060
   remote-address ipv4 200.200.200.0/24
   fast-register disable
   signaling-peer 200.200.200.98
   signaling-peer-port 5060
   account COREvlan
   registration target address 200.200.200.98
   registration target port 5060
   attach
```

SIP UAS Failure Detection

A User Agent Server (UAS) is a logical entity that generates a response to a SIP request. UAS failure detection is used to periodically monitor the state of a SIP network entity specified as the signaling peer on a SIP adjacency. SIP OPTIONS messages are sent to these network entities as a ping mechanism and a response from the device is expected. If a response is not received from the device, it is considered unreachable and removed from the routing calculations. Calls which cannot be routed through an alternate device are immediately responded to with a 604 Does Not Exist Anywhere message.

Cisco Unified Border Element (SP Edition) by default acts as an UAS that responds to OPTION pings when OPTION pings are sent to it. SIP UAS Failure Detection enables Cisco Unified Border Element (SP Edition) to send a SIP OPTIONS message to the device specified in the SIP Adjacency Destination Address. If an acceptable response is received within the SIP transaction timeout period then the routing tables are updated and the device is considered routable.

A ping failure occurs when no acceptable response is received within the SIP transaction timeout period. If ping-fail-count failures occur, then the device is considered to be unreachable. The signaling peer is considered offline as far as routing is concerned. Cisco Unified Border Element (SP Edition) sends pings at the rate specified in the period.

Note When the SBC has a TCP-based adjacency with OPTION ping enabled and that adjacency does not have a valid peer with which a TCP connection can be established, then that adjacency must be in the “no attach” state. This prevents the SBC from attempting to set up a TCP connection to a non-existent peer to send an OPTIONS ping message.

Use the procedure in this section to configure SIP UAS Failure Detection:

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **ping-enable**
6. **ping-interval** ***interval***
7. **ping-lifetime** ***duration***
8. **ping-fail-count** ***fail-count***
9. **exit**

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 *service-name* argument to define the name of the service.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

	Command or Action	Purpose
Step 4	**adjacency sip** *adjacency-name* **Example:** `Router(config-sbc-sbe)# adjacency sip sipGW`	Enters the mode of an SBE SIP adjacency. Use the *adjacency-name* argument to define the name of the service.
Step 5	**ping-enable** **Example:** `Router(config-sbc-sbe-adj-sip)# ping-enable`	Configures the adjacency to poll its remote peer by sending SIP OPTIONS pings to it and enters the ping option submode.
Step 6	**ping-interval** *interval* **Example:** `Router(config-sbc-sbe-adj-sip-ping)# ping-interval 100`	Configures the interval between SIP OPTIONS pings sent to the remote peer.
Step 7	**ping-lifetime** *duration* **Example:** `Router(config-sbc-sbe-adj-sip-ping)# ping-lifetime 100`	Configures the duration for which SBC waits for a response to an options ping for the adjacency.
Step 8	**ping-fail-count** *fail-count* **Example:** `Router(config-sbc-sbe-adj-sip-ping)# ping-fail-count 10`	Configures the number of consecutive pings that must fail before the adjacencies peer is deemed to be unavailable.
Step 9	**exit** **Example:** `Router(config-sbc-sbe-adj-sip)# exit`	Exits **adj-sip** mode to **sbe** mode.

SIP UAS Failure Detection: Example

In the following configuration example, PING is enabled on each of three adjacencies. A round robin call policy is set so that calls are distributed between the three adjacencies in a weighted random manner. If a UAS is unreachable, calls will be distributed between the remaining two adjacencies.

```
sbc mySBC
  sbe
    adjacency sip CallMgrA
      signaling-address ipv4 88.103.29.100
      remote-address ipv4 200.200.200.0 255.255.255.0
      signaling-peer 200.200.200.118
      ping-enable
        ping-interval 5
        ping-fail-count 3
        ping-lifetime 32
      attach

    adjacency sip CallMgrB
      signaling-address ipv4 88.103.29.100
      remote-address ipv4 200.200.200.0 255.255.255.0
      signaling-peer 200.200.200.200.117
      ping-enable
        ping-interval 5
        ping-fail-count 3
        ping-lifetime 32
      attach

    adjacency sip CallMgrC
      signaling-address ipv4 88.103.29.100
      remote-address ipv4 200.200.200.0 255.255.255.0
      signaling-peer 200.200.200.200.115
      ping-enable
        ping-interval 5
        ping-fail-count 3
        ping-lifetime 32
      attach

call-policy-set 1
      first-call-routing-table DestAddr
      rtg-dst-address-table DestAddr
        entry 1
          action next-table RoundRobin
          match-address 12
          prefix
      rtg-round-robin-table RoundRobin
        entry 1
          action complete
          dst-adjacency CallMgrB
        entry 2
          action complete
          dst-adjacency CallMgrC
        entry 3
          action complete
          dst-adjacency CallMgrA
      complete
    active-call-policy-set 1
```

SIP Outbound Flood Protection

SIP Outbound Flood Protection protects other network elements from excessively high valid traffic in unusual situations, such as a protection from a flood of generated BYE messages when a neighboring network element fails.

SIP Outbound Flood Protection sets a maximum rate of outgoing request messages and prevents the rate of outgoing request messages exceeding this maximum rate. If the limit is reached, outgoing requests are failed or dropped instead.

SIP Outbound Flood Protection is an addition to the normal CAC policy mechanisms and does not replace CAC policy. CAC policy allows fine grain control of calls, like, for example, rate limiting of INVITE requests at configurable scopes. SIP Outbound Flood Protection is intended to provide a simple overall rate limit for outgoing requests and is especially useful for requests that currently do not involve CAC policy (such as BYE requests).

Flood protection may be required in the following situations:

- Adjacent network element terminating — If an adjacent network element terminates (either normally or due to error) Cisco Unified Border Element (SP Edition) is likely to detect that the calls that used this element are dead at approximately the same time and attempt to tear the calls down. With many active calls this can generate a flood of BYE requests (normally two BYEs for each call).

Rather than allow these BYE messages to transiently overload other network signaling elements the network administrator may prefer to drop or fail some BYE requests at the Cisco Unified Border Element (SP Edition).

- Local removal of configuration in the Cisco Unified Border Element (SP Edition) — If a SIP adjacency is unconfigured using normal deactivation mode then BYE requests will be sent for all active calls using the adjacency before they are destroyed.

  Again it may be desirable for to limit the rate of outgoing requests prevent other network elements getting overloaded.

Use the procedure in this section to configure SIP Outbound Flood Protection:

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **outbound-flood-rate** ***rate***
6. **exit**

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 *service-name* argument to define the name of the service.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	**`adjacency sip`** *`adjacency-name`*
	Example:
	`Router(config-sbc-sbe)# adjacency sip sipGW`
		Enters the mode of an SBE SIP adjacency.
		Use the *adjacency-name* argument to define the name of the service.

	Command or Action	Purpose

Step 5	**`outbound-flood-rate`** ***`rate`***
	Example:
	`Router(config-sbc-sbe-adj-sip)# outbound-flood-rate 1000`
		Configures the maximum desired rate of outbound request signals on this adjacency (excluding ACK/PRACK requests) in signals per second.

Step 6	**`exit`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# exit`
		Exits **adj-sip** mode to **sbe** mode.

SIP Outbound Flood Protection: Example

The following configuration example sets an outbound flood rate of 100 outbound request signals per second.

```
sbc mySBC
  sbe
    adjacency sip CallMgrA
      signaling-address ipv4 88.103.29.100
      remote-address ipv4 200.200.200.0 255.255.255.0
      signaling-peer 200.200.200.118
      outbound-flood rate 100
      attach
```

SIP Over TLS

This section describes the concepts for SIP over Transport Layer Security (TLS). This section contains the following topics:

Security Configuration on an Adjacency

You can independently configure client and server security support on a SIP adjacency, using the following options:

- Untrusted—Specifies that this adjacency is not secured by any means. Only unsecured calls (not the calls to SIPS URIs) are made out of this adjacency.
- Untrusted-Encrypted—Specifies that the adjacency is untrusted and SSL/TLS encryption is used.
- Trusted-Encrypted—Specifies that the encrypted signaling is used to ensure security on this adjacency. The default certificate and key of the router are used for encryption. Only secure calls (calls to SIPS URIs) are made out of this adjacency.
- Trusted-Unencrypted—Specifies that a non-encryption mechanism is used to guarantee secure signaling for all messages on this adjacency. For example, this mechanism could be a single trusted physical link. Either secure or unsecured calls are made out of this adjacency. This configuration allows endpoints that do not support encryption to participate in secure SIP calls.

SIP Over TLS Overview

SIP Over Transport Layer Security (TLS) encryption provides a secure, encrypted transport to carry all SIP messages from the caller to the callee's domain. From there, the request is sent securely to the callee.

Cisco Unified Border Element (SP Edition) provides the following support for SIP Over TLS:

- Secured SIP calls can flow through Cisco Unified Border Element (SP Edition).
- A SIP adjacency can be secured by encryption or by another mechanism (for example, a single trusted physical-layer link or an interface to a trusted network).
- Inbound and outbound connections are immediately closed if a remote peer attempts to use encryption when encryption is not supported.
- Inbound and outbound connections are immediately closed if a remote peer fails to use encryption when encryption is required.
- You can view the level of security support configured for a given SIP adjacency by using the **show sbc** *sbc-name* **sbe adjacencies** *adj name* **detail** command.
- Calls received on untrusted adjacencies are not routed over outbound secure-encrypted adjacencies.
- Adjacencies secured by means of encryption can listen by default on port 5061. The port is configured to a different value.
- The fully-qualified domain name (FQDN) in the certificate offered by the remote peer is checked against the domain from which the request is received. The signal is dropped if the two do not match.
- Advanced Encryption Standard (AES) 128-bit Secure Hash Algorithm (SHA) is supported.
- The PKI HA updates the standby router with the certificate and trustpoint configuration changes.

The following are main security factors that are used in routing or rejecting a call:

- Calls to a SIPS URI must be secure. Calls to a SIP URI do not have to be secure.
- Signals received on a trusted adjacency are considered secure. Signals received on an untrusted adjacency are considered unsecured.

The following security factors apply to untrusted encrypted adjacencies:

- Secure calls may not be received on untrusted adjacencies of any type.
  - Cisco Unified Border Element (SP Edition) allows unsecured calls to be received over the untrusted encrypted adjacency.
  - Cisco Unified Border Element (SP Edition) rejects secured call that it receives over the untrusted encrypted adjacency.
- Secure calls cannot be routed to untrusted adjacencies.
  - Cisco Unified Border Element (SP Edition) can route unsecured calls over the untrusted encrypted adjacency.
  - Cisco Unified Border Element (SP Edition) does not route secured calls over the untrusted encrypted adjacency.

Table 6-1 and Table 6-2 summarize how Cisco Unified Border Element (SP Edition) handles inbound and outbound calls based on the call type, trust relationship, and encryption.

Table 6-1 Inbound Call Policy

	Trusted Adjacency		**Untrusted Adjacency**	
SIP Call Type	**Encrypted**	**Unencrypted**	**Encrypted**	**Unencrypted**
Secure SIP	Allow	Allow	Reject	Reject
Unsecured SIP	Allow	Allow	Allow	Allow

Table 6-2 Outbound Call Policy

	Trusted Adjacency		**Untrusted Adjacency**	
SIP Call Type	**Encrypted**	**Unencrypted**	**Encrypted**	**Unencrypted**
Secure SIP	Allow	Allow	Reject	Reject
Unsecured SIP	Allow	Allow	Allow	Allow

Table 6-3 and Table 6-4 summarize how Cisco Unified Border Element (SP Edition) handles inbound and outbound registrations based on the registration type, trust relationship, and encryption.

Table 6-3 Inbound Registration Policy

SIP Registration Type	**Trusted Adjacency**		**Untrusted Adjacency**	
	Encrypted	**Unencrypted**	**Encrypted**	**Unencrypted**
Secure SIP	Allow	Allow	Reject	Reject
Unsecured SIP	Allow	Allow	Allow	Allow

Table 6-4 Outbound Registration Policy

SIP Registration Type	Trusted Adjacency		Untrusted Adjacency	
	Encrypted	**Unencrypted**	**Encrypted**	**Unencrypted**
Secure SIP	Allow	Allow or Reject (depending on the configuration)	Reject	Reject
Unsecured SIP	Allow	Allow	Allow	Allow

For the SBC to be able to forward Secure SIP (SIPS) registrations to a trusted-unencrypted adjacency, all the following conditions must be met:

- The source adjacency must have a non-IP Multimedia Subsystem (non-IMS) or non-IMS access adjacency profile that specifies tracking of the registration state.
- The destination adjacency must have a non-IMS adjacency profile.
- The destination adjacency must be configured to accept a SIPS URI registration. This procedure is explained in the ?$paranum>Enabling the Conversion of SIPS URIs to SIP URIs on a Trusted-Unencrypted Adjacency? section on page 6-30.

When the SBC forwards secure registrations to a trusted-unencrypted adjacency that meets these conditions, the outbound registration is modified as follows:

- The Address of Record (AoR) in the To and From headers is converted from a SIPS URI to a SIP URI.
- The Request URI is converted from a SIPS URI to a SIP URI. Note that the Request URI may not be identical to the AoR.
- The URIs in the Contact headers are converted from SIPS to SIP.
- The URIs in Record Route headers are passed through unchanged. Note that according to RFC 3261, Record Route headers must be ignored on receipt if they are present in REGISTER messages.
- The URIs in other SIP headers are passed through unchanged.

Note The SBC rejects registrations that contain a mix of SIP URIs and SIPS URIs in their AoRs and contacts. On receipt of the REGISTER response, the SBC reverses the changes and passes back SIPS URIs in the response forwarded to the endpoint.

The following are restrictions for this feature:

- There is no change in the processing of non-INVITE messages, such as SUBSCRIBE, NOTIFY, and PUBLISH, by the SBC. For these messages, the SBC does not convert SIPS URIs to SIP URIs.
- The SBC does not support registrations to trusted-unencrypted adjacencies in scenarios where either the inbound adjacency or the outbound adjacency has an IMS profile.

User Agent Server-Side Processing

Inbound requests are marked according to two factors: whether the caller is trusted, and whether the call is intended for a secure target.

The caller-trust is determined in the following ways:

- SIP requests arriving over trusted adjacencies are marked as trusted.
- SIP requests arriving over untrusted adjacencies are marked as untrusted.

Desired target security is determined in the following ways:

- Requests for SIPS URIs are marked to require the outbound-security.
- Requests for SIP URIs are marked not to require the outbound-security.

Inbound requests are rejected if the caller is untrusted and the target requires security. All other combinations are forwarded to routing processing.

Routing Processing

The Routing Policy System (RPS) policy determines where the requests are routed next, with the following default behaviors:

- If a call requires the outbound security, the RPS considers only the trusted outbound adjacencies.
- If a call does not require the outbound security, the RPS considers only untrusted or trusted-unencrypted outbound adjacencies.

If the RPS is unable to find a suitable outbound adjacency for a call, the call is rejected.

User Agent Client-Side Processing

Outbound adjacencies preserve the URI scheme of the original request, ensuring that if a call is originally targeted at a SIPS URI, it is sent out to a SIPS URI. Or, if the call is originally targeted at a SIP URI, it is sent out to a SIP URI.

Upon receipt of 3xx class responses and target-refresh indications, the contact set is examined. Untrusted adjacencies do not permit the target of the call to be rerouted to a SIPS target. Likewise, trusted adjacencies do not permit the target of the call to be rerouted to a SIP target. If this is attempted by the remote peer, the call is brought down.

Configurable Mutual TLS Authentication Per Interface

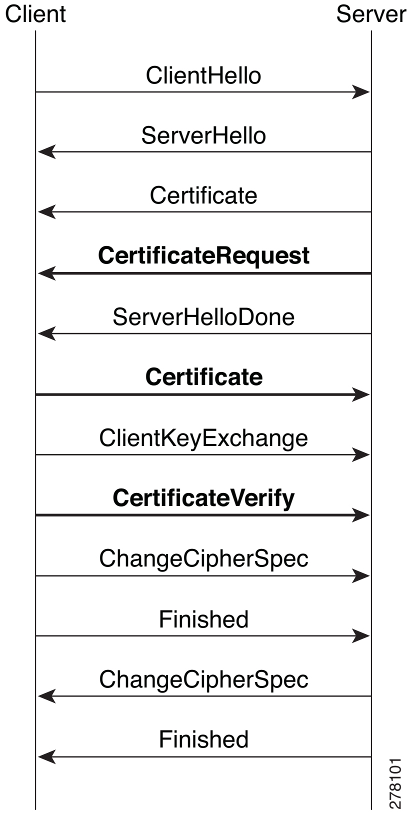
The Configurable Mutual TLS Authentication Per Interface feature helps you to configure unilateral or mutual TLS authentication on a per adjacency basis for SIP over TLS calls.

In a SIP over TLS call, SBC can either be a TLS client side or TLS server side. This feature is relevant only when the SBC is a TLS server side.

While negotiating a TLS connection, the server side sends its certificate to the client side to perform the server authentication. After the server authentication, the server may require a certificate from the client for client authentication. When client authentication is not used, the authentication is referred to as a unilateral authentication. When both — server and client — requires authentication, then it is referred to as a mutual authentication.

The message flow diagram, Figure 6-5, illustrates the negotiation process of a TLS connection. The bold line represents the messages required when mutual authentication is enabled on the server side:

Figure 6-5 Mutual TLS Authentication Message Flow Diagram



When SBC acts as TLS client side, it can automatically negotiate with the server side to perform the client authentication. But when SBC acts as TLS server side, you need to configure SBC so that SBC can decide whether to send a CertificateRequest message to the client side to get the client's certificate to do client authentication.

Use the **tls mutual-authentication** command to configure mutual authentication.

Restrictions and Limitations—Configurable Mutual TLS Authentication

- The configuration on a SIP adjacency cannot be modified while the adjacency is attached.
- The security configuration of the adjacency must be trusted encrypted or untrusted encrypted.
- Multiple TLS-enabled adjacencies that use the same local address and port must have the same configuration. Otherwise, the configuration will be rejected and an error message will be displaced on the console.
- Configuring trust points on a per adjacency basis is not possible because SBC uses global trust points to validate peer's certification. This limitation will not pose any limitation for certificate verification on SBC because, SBC automatically searches for a matching certificate from its global trust points.

- SBC only supports one certificate while SBC is a TLS server side. It is not possible to configure different certificates for each adjacency. The certificate is picked from the primary trust point.
- Certificate chain is not synchronized in SSO config-sync redundancy mode and hence the TLS certificates are not replicated to the standby. The incoming TLS calls might fail because of non-availability of TLS certificates.

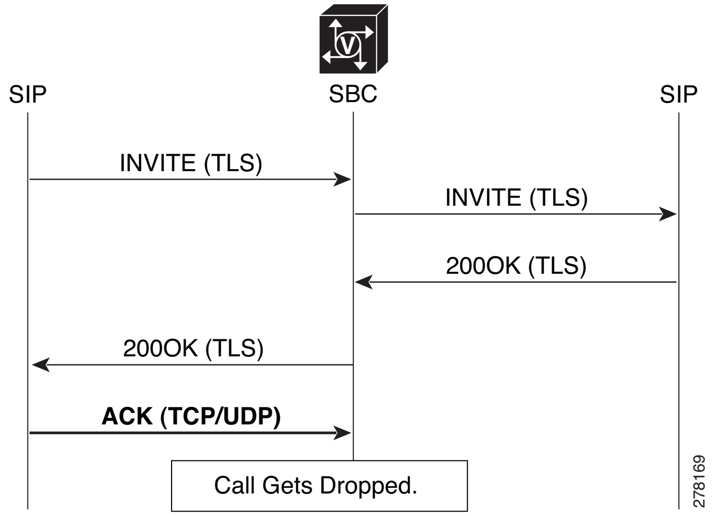
TLS Transport Parameter in Record-Route Headers

This feature allows you to add a **transport=tls** parameter to the SBC-originated Contact and Record-Route headers when using TLS. This feature is applicable when the security for the SBC inbound adjacency is configured as untrusted-encrypted.

The transport=tls parameter was deprecated in RFC3261 for better interoperability. With the implementation of RFC3261, the Contact and Record-Route header of 200(INVITE), back to caller, would use SIP URI without the transport parameter such as *Contact: <sip:192.168.1.1:5060>, Record-Route: <sip:192.168.1.1:5060;lr>*. Because of this, the subsequent mid-dialog requests—re-INVITE—are sent using TCP or UDP based on the SIP URI instead of TLS. Since SBC is expecting a TLS message on the port, the call is dropped.

Figure 6-6 shows the message flow where the SIP call is received over TLS, but the call was dropped. The ACK in response to the 200OK (TLS) message is sent from the SIP to SBC using TCP or UDP.

Figure 6-6 Message Flow During a SIP Call Over TLS



To avoid call drops, the caller is forced to use the TLS transport for the ACK by adding the transport=tls parameter. This feature is controlled on a per adjacency basis.

Use **header-name [contact [add [tls-param]] | from{passthrough} | to{passthrough}] command in (config-sbc-sbe-adj-sip) mode to configure the transport=tls parameter in the** Contact and Record-Route header.

Configuring SIP Over TLS on Cisco Unified Border Element (SP Edition)

Use the procedure in this section to configure SIP over TLS on Cisco Unified Border Element (SP Edition):

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **security trusted-encrypted**
6. **redirect-mode {pass-through | recurse}**
7. authentication nonce
8. **signaling-address ipv4** *ipv4_IP_address*
9. **signaling-port** *port-num*
10. remote-address ipv4 *ip-address ip-mask*
11. **signaling-peer** *peer-name*
12. **signaling-peer-port** *port-num*
13. **dbe-location-id** *dbe-location-id*
14. **reg-min-expiry** *period*
15. **attach** *force* **[abort | normal]**

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 *service-name* argument to define the name of the service.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	**`adjacency sip`** *`adjacency-name`*
	Example:
	`Router(config-sbc-sbe)# adjacency sip sipGW`
		Enters the mode of an SBE SIP adjacency.
		Use the *adjacency-name* argument to define the name of the service.

	Command or Action	Purpose
Step 5	**security trusted-encrypted** **Example:** `Router(config-sbc-sbe-adj-sip)# security trusted-encrypted`	Configures transport-level security on a Session Initiation Protocol (SIP) adjacency.

Step 6	**redirect-mode {pass-through | recurse}**
	Example:
	`Router(config-sbc-sbe-adj-sip)# redirect-mode recurse`
		Configures the behavior of SBC on receipt of a 3xx response to an invite from the SIP adjacency.

Step 7	**authentication nonce timeout** **Example:** `Router(config-sbe-adj-sip)# authentication nonce timeout 10`	Configures the authentication nonce timeout for a SIP adjacency.
Step 8	**signaling-address ipv4** ***ipv4_IP_address*** **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 10.10.10.10`	Defines the local IPv4 signaling address of a SIP or an H.323 adjacency.
Step 9	**signaling-port** ***port-num*** **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-port 5000`	Defines the local port of signaling address of a SIP adjacency.
Step 10	**remote-address ipv4** ***ip-address ip-mask*** **Example:** `Router(config-sbc-sbe-adj-sip)# remote-address ipv4 36.36.36.20 255.255.255.0`	Configures a SIP adjacency to restrict the set of remote signaling peers that can be contacted over the adjacency to those with the given IP address prefix.
Step 11	**signaling-peer peer-name** **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-peer 10.1.2.3`	Configures a SIP adjacency to use the given remote signaling-peer.
Step 12	**signaling-peer-port** ***port-num*** **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-peer-port 123`	Configures a SIP adjacency to use the given remote signaling-peer's port.
Step 13	**dbe-location-id** ***dbe-location-id*** **Example:** `Router(config-sbc-sbe-adj-sip)# dbe-location-id 1`	Configures an adjacency to use a given media gateway DBE location when routing media.

	Command or Action	Purpose
Step 14	**reg-min-expiry** ***period*** **Example:** Router(config-sbc-sbe-adj-sip)# reg-min-expiry 300	Configures the minimum registration period in seconds on the SIP adjacency.
Step 15	**attach** ***force*** **[abort | normal]** **Example:** Router(config-sbc-sbe-adj-h323)# attach	Attaches an adjacency to an account on an SBE.

SIP Over TLS Configuration Example

The following example shows a SIP over TLS configuration:

```
crypto pki trustpoint CA
 enrollment terminal
 serial-number
 subject-name ST=Some-State, C=AU, O=Internet Widgits Pty Ltd  revocation-check none
rsakeypair the_default !
!
crypto pki certificate chain CA
 certificate 01
  308201D7 30820140 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  46310B30 09060355 04061302 5553310C 300A0603 55040813 03525450 310B3009
  06035504 07130253 4A310E30 0C060355 040A1305 43495343 4F310C30 0A060355
  040B1303 53424330 1E170D30 39303230 35313030 3832385A 170D3130 30323035
  31303038 32385A30 45311330 11060355 0408130A 536F6D65 2D537461 7465310B
  30090603 55040613 02415531 21301F06 0355040A 1318496E 7465726E 65742057
  69646769 74732050 7479204C 7464305C 300D0609 2A864886 F70D0101 01050003
  4B003048 024100DC 18647810 B82F521B 40762B30 31646EB1 D567F0A6 E38DAD77
  1C41D825 E5274FFC A1F59E98 DCDFA617 161EA4D4 DBDC06E9 E1142752 9212D34D
  646E6B37 99D26502 03010001 A31A3018 30090603 551D1304 02300030 0B060355
  1D0F0404 030205A0 300D0609 2A864886 F70D0101 04050003 81810084 7E9A479B
  018F93F0 E683AA41 D3303705 6D89D44B 7798BD5F 15BCFAD5 EF55D72E 03365CD9
  BBCD955E 3C6D78B3 8E8C0675 772A7DE2 BCFBD6DF 760F9683 F0AB6F62 A87D9AC1
  AB2EA7E0 D831D33D 2F54582F 9E39F81D CBA33BD9 2466296C 4DCDAD0C 7D697AF7
  797AFEAA 05C3021F A7E89044 EA1796DC F422C82E 2B3894F6 3B98A7
        quit
 certificate ca 00F2D75C678DC7F7F2
  3082021A 30820183 A0030201 02020900 F2D75C67 8DC7F7F2 300D0609 2A864886
  F70D0101 04050030 46310B30 09060355 04061302 5553310C 300A0603 55040813
  03525450 310B3009 06035504 07130253 4A310E30 0C060355 040A1305 43495343
  4F310C30 0A060355 040B1303 53424330 1E170D30 39303230 35303931 3032395A
  170D3134 30323034 30393130 32395A30 46310B30 09060355 04061302 5553310C
  300A0603 55040813 03525450 310B3009 06035504 07130253 4A310E30 0C060355
  040A1305 43495343 4F310C30 0A060355 040B1303 53424330 819F300D 06092A86
  4886F70D 01010105 0003818D 00308189 02818100 BD3DBEEE A8CB6C51 9E2BBEC4
  35C2644F 92055B30 3543CA9D A1E1C0CB F59A2490 9296304D 43C19913 2A12EA80
  BDC6A1E3 0C164059 2C0DF132 E4AFF260 E88F38DC F23E866C DAFDD1BD F888BE90
  B74C49DA 4712E1E2 E249F444 FB3226B2 A5963DCD E75467B3 83669794 13BB8E7B
  CAFE3830 85091839 9658999B C72395E1 07AB35D1 02030100 01A31030 0E300C06
  03551D13 04053003 0101FF30 0D06092A 864886F7 0D010104 05000381 8100A7E5
  662FDE66 01FC63BA 399D1D17 0336C35B F9D9AEAF 87DA9E05 6AD13B90 D11CB984
  9B90FF8E 123F03B3 4E035D6B AC79D399 FF92A09C 2E62B759 E716D1D5 ABA46796
  41BB570F 96B7EE47 FB779AD4 0C8790FC 15FC65D6 47F60BE4 EB498B63 6DC2FBD3
  9DD51D82 0EB80125 D5A8F71B F7B61A63 5B601A6D FEFCAEB6 B33BF38B 9A10
        quit
```

The Cisco Unified Border Element (SP Edition) configuration example is illustrated here.

```
Router# configure
Router(config)# sbc sbc-3
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# security trusted-encrypted
Router(config-sbc-sbe-adj-sip)# redirect-mode pass-through
Router(config-sbc-sbe-adj-sip)# authentication nonce timeout 300
Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 10.130.10.25
Router(config-sbc-sbe-adj-sip)# signaling-port 5060
Router(config-sbc-sbe-adj-sip)# remote-address ipv4 10.74.49.145 255.255.255.255
Router(config-sbc-sbe-adj-sip)# signaling-peer 10.74.49.145
Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060
Router(config-sbc-sbe-adj-sip)# dbe-location-id 4294967295
Router(config-sbc-sbe-adj-sip)# reg-min-expiry 3000
Router(config-sbc-sbe-adj-h323)# attach
```

SIP Over TLS Verification

Use the following commands to check certificates on the node:

show crypto pki certificates—displays information about your certificate, the certification authority certificate (CA), and any registration authority (RA) certificates.

show crypto key pubkey-chain rsa—enters public key configuration mode (so you can manually specify and show other devices' RSA public keys).

show crypto key mypubkey rsa—displays the RSA public keys of your router.

Enabling the Conversion of SIPS URIs to SIP URIs on a Trusted-Unencrypted Adjacency

Use the procedure described in this section to enable the conversion of SIPS URIs to SIP URIs on a trusted-unencrypted adjacency. Performing this procedure is one of the requirements for configuring the SBC to forward secure registrations to a trusted-unencrypted adjacency. See ?$paranum>SIP Over TLS Overview? section on page 6-21 for more information about this feature.

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *sbc-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **security trusted-unencrypted**
6. **registration unencrypted-convert**
7. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enables the global configuration mode.

Step 2	**`sbc`** *`sbc-name`*
	Example:
	`Router(config)# sbc mysbc`
		Enters the SBC service mode.
		- *sbc-name*—Name of the SBC.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the SBE configuration mode.

Step 4	**`adjacency sip`** *`adjacency-name`*
	Example:
	`Router(config-sbc-sbe)# adjacency sip sipGW`
		Enters the mode of an SBE SIP adjacency.
		adjacency-name—Name of the adjacency.

Step 5	**`security trusted-unencrypted`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# security trusted-encrypted`
		Configures transport-level security on the adjacency.

Step 6	**`registration unencrypted-convert`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# registration unencrypted-convert`
		Enables the conversion of SIPS URIs to SIP URIs on the trusted-unencrypted adjacency.

Step 7	**`end`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# end`
		Returns to the privileged EXEC mode.

In the following example, the output of the **show sbc adjacencies** command shows that conversion of SIPS URIs to SIP URIs is enabled:

```
Router# show sbc MySBC sbe adjacencies ADJ1 detail
SBC Service MySBC
  Adjacency ADJ1 (SIP)
    Status:                Attached
    Signaling address:    192.0.2.36.1:5060, VRF sidd_sipp1
    IPsec server port:    0
    Signaling-peer:       192.0.2.37.1:5060 (Default)
.
.
.

      Media Bypass Tag List:
      Tag 1:                     tag1
Tag 2:                     tag2
Media Bypass Max Out Data Length:        1024
Register unencrypted covert: Enabled
```

SIP Peer Availability Detection

The SBC supports the SIP peer availability detection (OPTIONs ping) functionality. The SBC periodically sends an OPTION request to a configured peer. If the peer fails to respond to a set number of OPTION requests, the peer is declared dead, and the calls are routed through other peers.

To avoid congestion, when ping suppression is enabled, and if signaling traffic exchange between peers is active, the OPTIONS pings are not used to check peer availability.

Restrictions for SIP Peer Availability Detection

The SIP Peer Availability Detection feature has the following restrictions:

- The OPTION requests will use the SIP method congestion response codes.
- If the number of OPTIONS messages to the peer are reduced, the time taken to detect dead peer by the SBC increases substantially.

Configuring SIP Peer Availability Detection

Use the procedure described in this section to configure the detection of SIP peer availability:

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **ping-enable**
6. **ping-bad-rsp-codes** *ranges*
7. **ping-suppression**
8. **exit**
9. **end**
10. **show sbc** *sbc-name* **sbe adjacencies** *adjacency-name* **Detail**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**`configure terminal`**
	Example:
	`Router# configure terminal`
		Enables the global configuration mode.

Step 2	**`sbc`** *`service-name`*
	Example:
	`Router(config)# sbc mysbc`
		Enters the mode of an SBC service.
		Use the *service-name* argument to define the name of the service.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	**`adjacency sip`** *`adjacency-name`*
	Example:
	`Router(config-sbc-sbe)# adjacency sip sipGW`
		Enters the mode of an SBE SIP adjacency.
		Use the *adjacency-name* argument to define the name of the service.

Step 5	**`ping-enable`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# ping-enable`
		Configures the adjacency to poll the adjacency's remote peer by sending SIP OPTION pings to it, and enters the Ping option submode.

Step 6	**`ping-bad-rsp-codes`** *`ranges`*
	Example:
	`Router(config-sbc-sbe-adj-sip-ping)#`
	`ping-bad-rsp-codes ranges 300,398`
		Configures the congestion response codes on a SIP adjacency by sending SIP OPTION pings to the adjacency.
		Use the *ranges* argument to specify the response code range (The range can be 300 to 399).

Step 7	**`ping-suppression`** ***`options`***
	Example:
	`Router(config-sbc-sbe-adj-sip-ping)#`
	`ping-suppression odd-request`
		(Optional) Configures the SBC to ping, when required, on a SIP adjacency.
		options specifies one of the following strings used for ping suppression:
		• ood-request—The SBC considers a peer as reachable when an out-of-dialog or dialog-creating request is received, excluding the OPTIONS and REGISTER messages.
		• ood-response—The SBC considers a peer as reachable when an out-of-dialog or dialog-creating 2xx response is received, excluding OPTIONS and REGISTER messages.
		• ind-request—The SBC considers a peer as reachable when an in-dialog request is received.
		• ind-response—The SBC considers a peer as reachable when an in-dialog 2xx response is received.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-sbc-sbe-adj-sip-ping)# exit`	Exits the adj-sip-ping mode, and moves to the adj-sip mode.

Step 9	**end**
	Example:
	`Router(config-sbc-sbe)# end`
		Exits the SBE mode and returns to the Privileged EXEC mode.

Step 10	**show sbc** *sbc-name* **sbe adjacencies** *adjacency-name* **detail** **Example:** `Router# show sbc mysbc sbe adjacencies sipGW detail`	Displays the details pertaining to the specified adjacency.

Example

The following example shows how to configure the congestion response codes on a SIP adjacency by sending SIP OPTIONS pings:

```
Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip SipAdj1
Router(config-sbc-sbe-adj-sip)# ping-enable
Router(config-sbc-sbe-adj-sip-ping)# ping-bad-rsp-codes ranges 300,398
Router(config-sbc-sbe-adj-sip-ping)# exit
Router(config-sbc-sbe-adj-sip)#
```

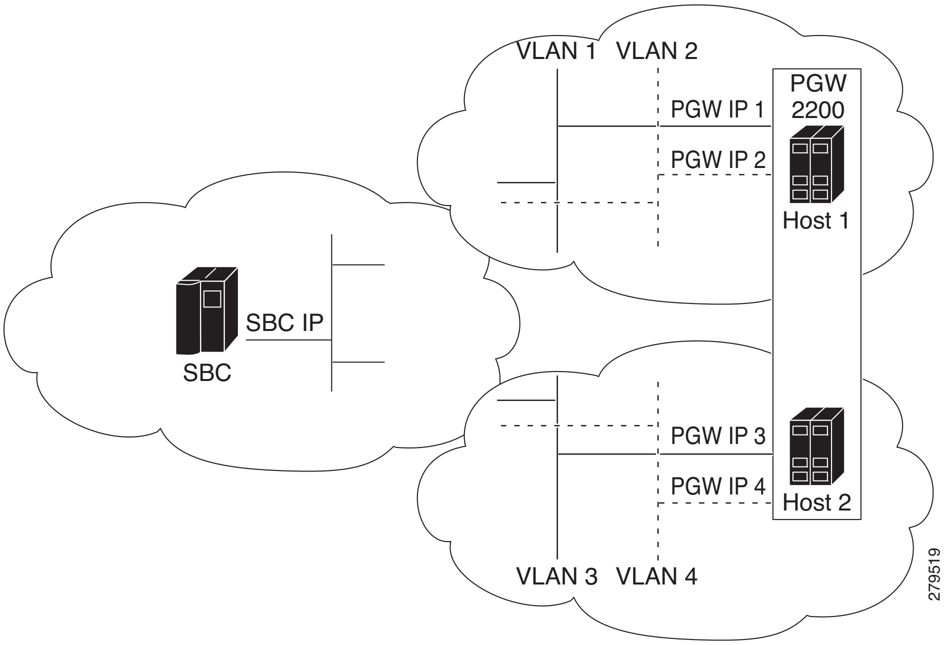
Redundant Peer Addresses

The SBC may be required to interoperate with peer SIP devices, such as the PGW 2200 softswitch, which can start signaling from a different IP address following a VLAN failure when operating with redundant hosts on separate VLANs (for example, geographically separated hosts). Peer SIP devices, such as the PGW 2200 softswitch, have the following high availability (HA) strategies that include both VLAN and host redundancy:

- In a standard failover scenario, when one host fails, a backup takes over. This backup also takes over the virtual IP address used for SIP communication. The call state is maintained, and the failover is made invisible to the SBC.
- In a scenario where a VLAN failure occurs, the PGW 2200 softswitch host, which interoperates with the SBC, starts using an interface in a different VLAN. Because you cannot transfer a virtual IP address between VLANs, the IP address for SIP communication changes.
- In some networks, the primary and backup hosts are geographically redundant and unable to share a VLAN. Therefore, you cannot transfer a virtual IP address between the hosts, so the IP address being used for SIP communication changes if the primary host fails.

Figure 6-7 illustrates a redundant topology.

Figure 6-7 Redundant Topology



When peer IP addresses change, the call state on the corresponding peer device continues to be maintained, and key SIP parameters, such as the dialog tags, the Contact header, and route set, are unchanged. However, the Contact header is incorrect, and does not contain the new peer IP address.

The Redundant Peer Addresses feature allows the IP address of its peer device to change, and supports the following functionalities on SBC:

- Accepts incoming SIP messages from any of the redundant IP addresses.
- Ignores discrepancies between the IP addresses specified in the SIP Contact header (or other SIP headers) and the actual IP address being used by the peer.
- Pings each peer address to monitor the active addresses and sends the outgoing SIP messages destined for the peer to an IP address that is currently active.
- Configures the SBC with multiple redundant IP addresses for a SIP peer device that is not contained within a single remote address mask.
- Supports the following modes of operation that is configurable for each adjacency:
  - The SBC switches peer IP address when a higher priority address becomes active, even if the current address does not fail.
  - Elects a *current destination* for each adjacency, choosing the peer IP address with the highest-priority that is currently active, and continues to use that destination until it is detected to have failed, at which point the election process is repeated.
- Uses a single local IP address, port, and VPN for all communication with every peer IP address.

Restrictions for Redundant Peer Addresses

The Redundant Peer Addresses feature has the following restrictions:

- This is a signaling-only feature.
- Alternative redundant addresses for a peer cannot be automatically detected, and must therefore, be configured using the **ping-enable** command.
- The main peer address in an adjacency share the same priority values, ranging from 1 to 6, as the redundant peer addresses.
- A single load-balancing method is provided. The SBC selects the active peer IP address with the highest configured priority for all the outgoing SIP requests.
- The source address of fast-REGISTER requests cannot be changed.
- If a SIP request is sent to a peer address, and no response is received, the SBC subsequently detects that the peer address has failed. However, the destination address of the SIP request does not change, and it is retried to the failed address. New requests are sent to an active address.
- The destination addresses and ports configured for a given adjacency are not available in message editing configuration. Therefore, there is no per-destination equivalent for the existing signaling-peer and signaling-peer-port header filtering syntax.
- The optimization to send only pings when required (ping suppression) cannot be configured on the adjacencies facing redundant peers.

Configuring Redundant Peer Addresses

Use the procedure described in this section to configure redundant peer addresses:

SUMMARY STEPS

1. **configure terminal**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **no attach**
6. **force-signaling-peer all**
7. **ping-enable**
8. **exit**
9. **redundant peer** *index*
10. **address** *address*
11. **port** *port*
12. **network** {**IPv4** *address netmask* | **IPv6** *address netmask*}
13. **priority** *priority*
14. **activate**
15. **exit**
16. **signaling-peer-switch** {**always** | **fail**}

17. **signaling-peer-priority** *priority*
18. **exit**
19. **end**
20. **show sbc** *sbc-name* **sbe adjacencies** *adjacency-name* **peers**

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 *service-name* argument to define the name of the service.

Step 3	**`sbe`**
	Example:
	`Router(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	**`adjacency sip`** *`adjacency-name`*
	Example:
	`Router(config-sbc-sbe)# adjacency sip sipGW`
		Enters the mode of an SBE SIP adjacency.
		Use the *adjacency-name* argument to define the name of the service.

Step 5	**`no attach`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# no attach`
		Detaches the adjacency from an account on the SBE.
		Note The adjacency must be detached before adding or removing a redundant peer.

Step 6	**`force-signaling-peer all`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# force-signaling-peer all`
		Forces SIP messages for both in-call requests and out-of-call requests to go to the configured signaling peer.

Step 7	**`ping-enable`**
	Example:
	`Router(config-sbc-sbe-adj-sip)# ping-enable`
		Configures the adjacency to poll its remote peer by sending SIP OPTIONS pings to it, and enters the ping option submode.

Step 8	**`exit`**
	Example:
	`Router(config-sbc-sbe-adj-sip-ping)# exit`
		Exits the **adj-sip-ping** mode, and moves to **adj-sip** mode.

	Command or Action	Purpose
Step 9	**redundant peer** *index* **Example:** `Router(config-sbc-sbe-adj-sip)# redundant peer 1`	Enters the mode of an SBE SIP adjacency peer to configure an alternative signaling peer for the adjacency. You can specify the index number of the peer, ranging from 1 to 5.
Step 10	**address** *address* **Example:** `Router(config-sbc-sbe-adj-sip-peer)# no address`	Configures either an IP address or a host name to act as the redundant peer.
Step 11	**port** *port* **Example:** `Router(config-sbe-adj-sip-peer)# port 2`	Configures a port for the redundant peer. **Note** By default, 5060 port is used.

Step 12	**network** {**IPv4** *address netmask* | **IPv6** *address netmask*}
	Example:
	`Router(config-sbc-sbe-adj-sip-peer)# network ipv4 33.33.36.2 255.255.255.0`
		Configures either an IPv4 or IPv6 network on the redundant peer.

Step 13	**priority** *priority* **Example:** `Router(config-sbc-sbe-adj-sip-peer)# priority 1`	Configures the redundant peer's priority. The range is from 1 to 6.
Step 14t	**activate** **Example:** `Router(config-sbc-sbe-adj-sip-peer)# activate`	Activates the redundant signaling peer.
Step 15t	**exit** **Example:** `Router(config-sbc-sbe-adj-sip-peer)# exit`	Exits the **adj-sip-peer** mode, and moves to **adj-sip** mode.

Step 16	**signaling-peer-switch** {**always** | **fail**}
	Example:
	`Router(config-sbc-sbe-adj-sip)# signaling-peer-switch always`
		Configure a SIP adjacency to switch the signaling peer to an available destination.

Step 17	**signaling-peer-priority** *priority* **Example:** `Router(config-sbc-sbe-adj-sip)# signaling-peer-priority 1`	Configures the priority of a signaling peer on a SIP adjacency. The range is from 1 to 6.
Step 18	**exit** **Example:** `Router(config-sbc-sbe-adj-sip)# exit`	Exits the **adj-sip** mode, and moves to **sbe** mode.

	Command or Action	Purpose

Step 19	**end**
	Example:
	`Router(config-sbc-sbe)# end`
		Exits the sbe mode and returns to privileged EXEC mode.

Step 20	**show sbc** *sbc-name* **sbe adjacencies** *adjacency-name* **peers** **Example:** `Router# show sbc mysbc sbe adjacencies sipGW peers`	Lists the configured peers for the specified adjacency.

Redundant Peer Addresses Example

The following example shows a redundant peer addresses configuration:

```
sbc mat
 sbe
   adjacency sip SIPPA
    force-signaling-peer all
    signaling-peer-switch on-fail
    inherit profile preset-access
    signaling-address ipv4 1.0.0.10
    statistics method summary
    signaling-port 5068
    remote-address ipv4 1.0.0.0 255.0.0.0
    signaling-peer 1.0.0.3
    signaling-peer-priority 6
    signaling-peer-port 5068
    registration rewrite-register
    registration target address 1.0.0.3
    registration target port 5068
    redundant peer 1
     network ipv4 5.5.5.5 255.255.255.255
     address 5.5.5.5
     priority 2
     activate
    redundant peer 2
     network ipv4 22.22.22.22 255.255.255.255
     address 22.22.22.22
     port 2222
     priority 3
    ping-enable
    attach
```

Redundant Peer Addresses Verification

Use the following commands to verify the peers:

- **show sbc sbe adjacencies detail**—Displays detailed configuration of a SIP adjacency.
- **show sbc sbe adjacencies peer—**Lists the configured peers for an adjacency.
- **show sbc sbe all-peers**—Displays a peer's information.