Implementing SBC Adjacencies

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 the Session Border Controller (SBC) will interact. Within each account, the user defines one or more signaling adjacencies, which connect the SBC 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 SBC 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.

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

For a complete description of commands used in this chapter, refer to Chapter 39, "Cisco Session Border Controller Commands.". To locate documentation for other commands that appear in this chapter, use the command reference master index, or search online.

Feature History for Implementing SBC Adjacencies

Release	Modification
ACE SBC Release 3.1.00	• Added support for improved fast register. This feature maintains registration data for each registered source address and examines inbound register messages to determine whether to handle the message locally or forward it to the SBC.
	• Added support for SIP Statistics Per Adjacency.
	• Added support for SIP PING Messages.

Contents

This module contains the following sections:

Prerequisites for Implementing Adjacencies

The following prerequisites are required to implement adjacencies:

- On the Application Control Engine Module (ACE), you must be an Admin user to enter SBC commands. For more information, see the Application Control Engine Module Administration Guide at http://www.cisco.com/en/US/docs/interfaces_modules/services_modules/ace/v3.00_A2/configuration/administration/guide/ace_adgd.pdf

- Before implementing adjacencies, the SBC must already be created. See the procedures described in Chapter 2, "ACE Configuration Prerequisites for the SBC".

Information About Implementing Adjacencies

Adjacencies are used to enable call signaling between the SBE and other voice over IP (VoIP) devices. The SBC supports adjacencies in both Session Initiation Protocol (SIP) and H.323 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.

- In an H.323 network, the devices might be terminals, gateways, or gatekeepers. When you configure an H.323 adjacency, the SBC functions as a gateway within the H.323 network.

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

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

- Properties Common to Both SIP and H.323 Adjacencies
- About SIP Adjacencies in the Deployment
- About H.323 Adjacencies in the Deployment
- How Adjacencies Affect Media Routing

Properties Common to Both SIP and H.323 Adjacencies

The following properties are common to both SIP and H.323 adjacencies:

- Adjacencies are known by name. The name makes it easy for an SBC policy to reference the adjacency.

- An adjacency has a local address and port for incoming call setup. The IP address must be an address that matches the VLAN interface alias.

- 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 11-1 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 11-1   SIP Network
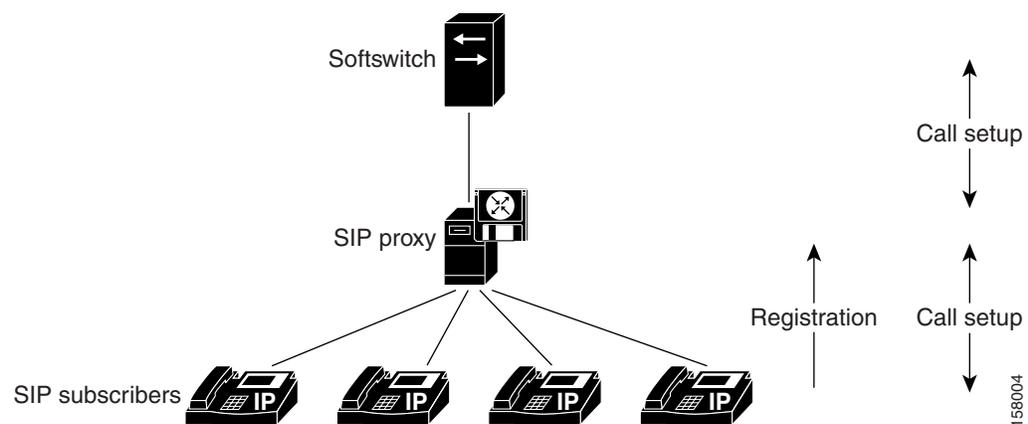
Figure 11-2 shows placement of an SBC in two possible positions within the SIP network, with the adjacencies noted. Each adjacency enables call setup to one or more neighboring devices, as follows:

- 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 SBC Policies module.

Figure 11-2   Adjacencies in a SIP Network Deployment
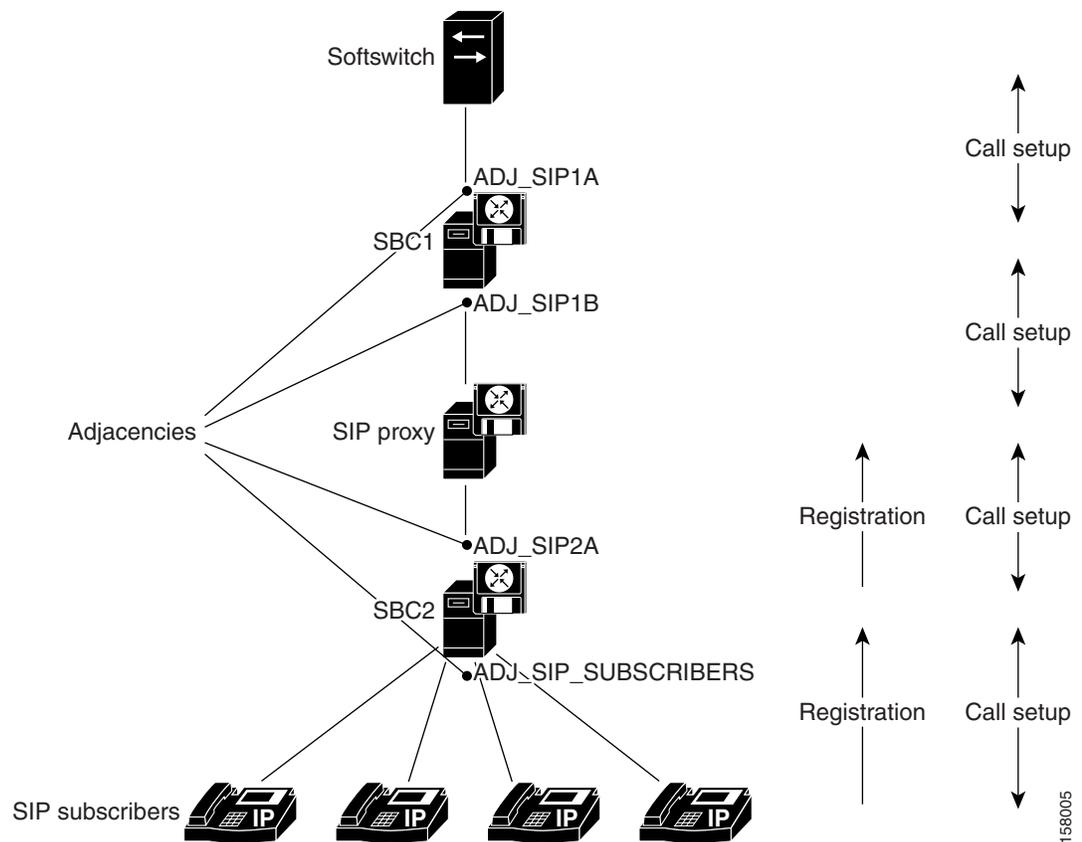
The SBC can operate in non-IMS networks using one of three different non-IMS profiles:

- An access adjacency faces user equipment, such as a subscriber's telephone, or other SIP device.

- A core adjacency links the access adjacency to the registrar.

- Peering adjacencies link one registrar to another.

By configuring each of these different types of adjacency with a profile, you can make efficiency and occupancy gains. For example, the SBC will not store registration information from messages received from peering adjacencies.

The differences between the above profiles against registrations and calls is as follows:

- Registrations—When a subscriber successfully registers from an access adjacency, the SBC remembers the subscriber's registration details for later use. SBC does not store this information on core or peering adjacencies.

- Calls—When an outgoing call is received from an endpoint on an access adjacency, the SBC checks to see if the subscriber is registered. If the subscriber is registered, the SBC applies subscriber policy to the call. Note that subscribers do not have to be registered to make an outgoing call.

  When a call is received on a core adjacency, the SBC checks to see if the endpoint is registered. If the endpoint is registered, the SBC can apply subscriber policy and route the call to the appropriate access adjacency. Additionally, if the registered subscriber is known to be behind a NAT, the SBC configures the call to traverse the NAT. If the endpoint is not registered, the SBC applies routing policy instead, and routes the call to the appropriate adjacency.

About H.323 Adjacencies in the Deployment

Figure 11-3 shows a simple H.323 network where:

- H.323 terminals and gateway both register with the gatekeeper.

- Prior to placing a call, an endpoint resolves the destination address with the gatekeeper.

- Call signaling flows directly between endpoints (whether terminals or gateway).

Figure 11-3   H.323 Network
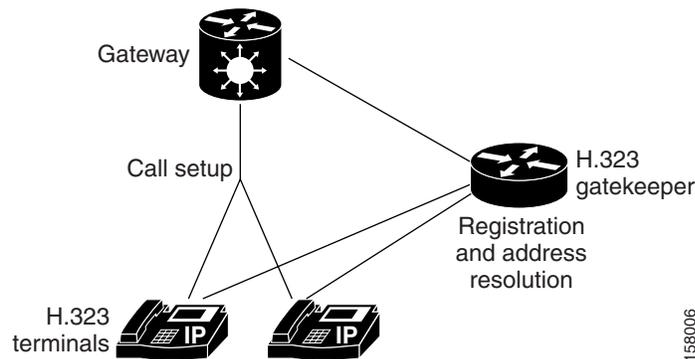
Figure 11-4 shows a possible deployment of an SBC in the network, with two adjacencies noted. Each adjacency allows call setup to one or more neighboring devices.

- ADJ_H3231A allows call setup between the SBC and the gateway.

- ADJ_H3231B allows call setup between the SBC and the terminals.

An H.323 adjacency may or may not be registered with a gatekeeper. In Figure 11-4, ADJ_H3231B is registered with a gatekeeper and ADJ_H3231A is not.

- Gatekeeper adjacencies can set up calls to multiple endpoints. Their peer address is set to the address of the gatekeeper.

- Non-gatekeeper adjacencies can set up calls to a single remote endpoint. Their peer address is set to that of the endpoint (for example, a gateway).

Figure 11-4   Adjacencies in an H.323 Network Deployment
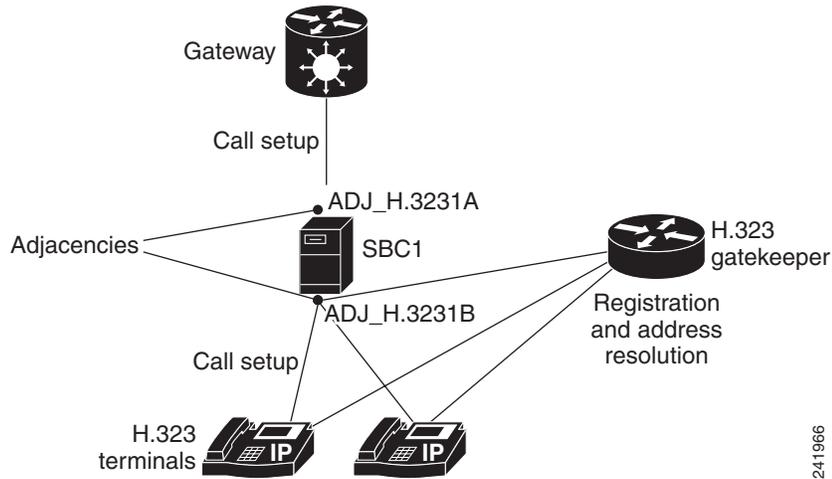
How Adjacencies Affect Media Routing

For a distributed SBC 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 SBC 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 11-5 and Figure 11-6 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 11-5   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 11-6.

- 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 11-6   Media Routing for Three Calls: Peer1-Peer3, Peer1-Peer2a, and Peer2a-Peer2b
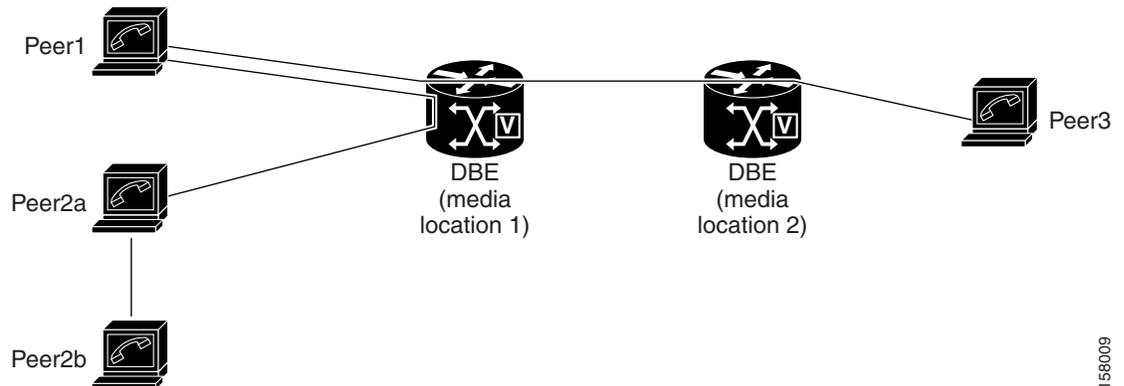
How to Implement Adjacencies

Adjacencies are the key objects used to control signaling. The user defines one or more signaling adjacencies, which connect the SBC 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 SBC 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 an H.323 adjacency and a SIP adjacency, depending on your implementation requirements:

- Configuring Force-Signaling-Peer Adjacency
- Assigning H.323 Adjacencies to Adjacency Groups
- Configuring a SIP Adjacency
- Assigning SIP Adjacencies to Adjacency Groups

Configuring Force-Signaling-Peer Adjacency

This task configures a force-signaling-peer adjacency.

SUMMARY STEPS

1. **configure**
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`
	
	Example:
	`host1/Admin# configure`
		Enables global configuration mode.

Step 2	`sbc` *service-name* **Example:** `host1/Admin(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:
	`host1/Admin(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	`adjacency sip` *adjacency-name* **Example:** `host1/Admin(config-sbc-sbe)# adjacency sip 2651XM-5`	Enters the mode of a SIP adjacency. Use the *adjacency-name* argument to define the name of the H.323 adjacency.

Step 5	`force-signaling-peer`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-sip)# force-signaling-peer`
		Forces SIP messages to go to the configured signaling peer.

Step 6	`attach`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-sip)# attach`
		Attaches the adjacency.

Step 7	`exit`
	
	Example:
	`host1/Admin(config-sbc-sbe-acc-ser)# exit`
		Exits the sip mode to the sbe mode.

Configuring an H.323 Adjacency

This task configures an H.323 adjacency.

SUMMARY STEPS

1. **configure**

2. **sbc** *service-name*

3. **sbe**

4. **adjacency h323** *adjacency-name*

5. **signaling-address ipv4** *ipv4_IP_address*

6. **signaling-port** *port_num*

7. **signaling-peer** [**gk**] *peer_address*

8. **signaling-peer-port** *port_num*

9. **remote-address ipv4** *ip-address ip-mask*

10. **account** *account_name*

11. **attach**

12. **exit**

13. **adjacency h323** *adjacency-name*

14. **signaling-address ipv4** *ipv4_IP_address*

15. **signaling-port** *port_num*

16. **signaling-peer** [**gk**] *peer_address*

17. **signaling-peer-port** *port_num*

18. **remote-address ipv4** *ipv4_IP_address/prefix*

19. **account** *account_name*

20. **show services sbc sbe adjacencies detail**

21. **attach**

22. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`host1/Admin# configure`
		Enables global configuration mode.

Step 2	`sbc service-name`
	
	Example:
	`host1/Admin(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:
	`host1/Admin(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	`adjacency h323 adjacency-name`
	
	Example:
	`host1/Admin(config-sbc-sbe)# adjacency h323`
	`2651XM-5`
		Enters the mode of an SBE H.323 adjacency.
		Use the *adjacency-name* argument to define the name of the H.323 adjacency.

	Command or Action	Purpose
Step 5	**signaling-address ipv4** *ipv4_IP_address* **Example:** host1/Admin(config-sbc-sbe-adj-h323)# signaling-address ipv4 88.88.137.10	Specifies the local IPv4 signaling address of the H.323 adjacency.
Step 6	**signaling-port** *port_num* **Example:** host1/Admin(config-sbc-sbe-adj-h323)# signaling-port 5000	Specifies the local signaling port of the H.323 adjacency.
Step 7	**signaling-peer** [**gk**] *peer_address* **Example:** host1/Admin(config-sbc-sbe-adj-h323)# signaling-peer 200.200.200.41	Specifies the remote signaling peer for the H.323 adjacency to use.
Step 8	**signaling-peer-port** *port_num* **Example:** host1/Admin(config-sbc-sbe-adj-h323)# signaling-peer-port 5000	Specifies the remote signaling-peer port for the H.323 adjacency to use.
Step 9	**remote-address ipv4** *ip-address ip-mask* **Example:** host1/Admin(config-sbc-sbe-adj-h323)# remote-address ipv4 1.1.1.1	Restricts the set of remote signaling peers contacted over the adjacency to those with the given IP address prefix.
Step 10	**account** *account_name* **Example:** host1/Admin(config-sbc-sbe-adj-h323)# account core-sided	Defines the H.323 adjacency as belonging to an account on an SBE.
Step 11	**attach** **Example:** host1/Admin(config-sbc-sbe-adj-h323)# attach	Attaches the adjacency.
Step 12	**exit** **Example:** host1/Admin(config-sbc-sbe-adj-h323)# exit	Exits the adj-h323 mode to the sbe mode.
Step 13	**adjacency h323** *adjacency-name* **Example:** host1/Admin(config-sbc-sbe)# adjacency h323 2651XM-6	Enters the mode of an SBE H.323 adjacency. Use the *adjacency-name* argument to define the name of the H.323 adjacency.

	Command or Action	Purpose

Step 14	`signaling-address ipv4 ipv4_IP_address`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# signaling-address ipv4 88.88.137.10`
		Specifies the local IPv4 signaling address of the H.323 adjacency.

Step 15	`signaling-port port_num`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# signaling-port 5050`
		Specifies the local signaling port of the H.323 adjacency.

Step 16	`signaling-peer [gk] peer_address`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# signaling-peer 10.10.119.12`
		Specifies the remote signaling peer for the H.323 adjacency to use.

Step 17	`signaling-peer-port port_num`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# signaling-peer-port 5050`
		Specifies the remote signaling-peer port for the H.323 adjacency to use.

Step 18	`remote-address ipv4 ipv4_IP_address/prefix`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# remote-address ipv4 10.10.l19.0/24`
		Restricts the set of remote signaling peers contacted over the adjacency to those with the given IP address prefix.

Step 19	`account account_name`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# account node-side`
		Defines the H.323 adjacency as belonging to an account on an SBE.

Step 20	`show services sbc sbe adjacencies detail`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# show services sbc sbc sbe adjacencies sip-shanghai detail`
		Shows all the fields specified SIP adjacency.

Step 21	`attach`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# attach`
		Attaches the adjacency.

Step 22	`exit`
	
	Example:
	`host1/Admin(config-sbc-sbe-acc-ser)# exit`
		Exits the adj-h323 mode to the sbe mode.

Assigning H.323 Adjacencies to Adjacency Groups

Use the procedure in this section to assign an H.323 adjacency to an adjacency group.

SUMMARY STEPS

1. **configure**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency h323** *adjacency-name*
5. **group** *adjacency-group-name*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`host1/Admin# configure`
		Enables global configuration mode.

Step 2	`sbc service-name`
	
	Example:
	`host1/Admin(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:
	`host1/Admin(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	`adjacency h323 adjacency-name`
	
	Example:
	`host1/Admin(config-sbc-sbe)# adjacency h323 2651XM-5`
		Enters the mode of an SBE H.323 adjacency.
		
		Use the *adjacency-name* argument to define the name of the H.323 adjacency.

Step 5	`group adjacency-group-name`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# group isp1`
		Assigns the H.323 adjacency to an adjacency group.
		
		Use the *adjacency-group-name* argument to define the group name.

Step 6	`exit`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-h323)# exit`
		Exits the adj-h323 mode to the sbe mode.

Configuring a SIP Adjacency

⚠️

Caution   Adjacencies can only be modified when the status is detached. Use the **no attach force** command to force the adjacency to go to the detached state.Use the **show services sbc adjacencies** command to check the state. To modify an adjacency, use the **no attach** command first. When a call is active or when the ping enable feature is running, the adjacency stays in the going down state. During this state, existing calls are not torn down and new calls are not accepted. The adjacency will not go to detached until all calls have ended. An adjacency cannot be attached until the adjacency goes to detached.

✎

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 a preset-core value. When using the default value for those adjacencies without specific per-adjacency configuration, the **sip inherit profile preset-core** 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-access** command on the adjacency facing subscribers and the **inherit profile preset-core** command on the adjacency facing the the SIP proxy. If you use other combinations (for example, if both the inbound and outbound adjacencies are configured as preset-core, the SBC does not store the registration information, nor does it rewrite the Contact: header to make sure that it is 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**

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. **show**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`host1/Admin# configure`
		Enables global configuration mode.

Step 2	`sbc service-name`
	
	Example:
	`host1/Admin(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:
	`host1/Admin(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:** `host1/Admin(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:
	`host1/Admin(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 6	**signaling-address ipv4** *ipv4_IP_address* **Example:** host1/Admin(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.88.141.3	Specifies the local IPv4 signaling address of the SIP adjacency.
Step 7	**signaling-port** *port_num* **Example:** host1/Admin(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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(config-sbc-sbe-adj-sip)# registration rewrite-register	Configures SIP REGISTER request rewriting.
Step 13	**attach** **Example:** host1/Admin(config-sbc-sbe-adj-sip)# attach	Attaches the adjacency.
Step 14	**exit** **Example:** host1/Admin(config-sbc-sbe-adj-sip)# exit	Exits **adj-sip** mode to **sbe** mode.

	Command or Action	Purpose
Step 15	**adjacency sip** *adjacency-name* **Example:** host1/Admin(config-sbc-sbe-adj-sip)# adjacency sip sipPROXY	Enters the mode of an SBE SIP adjacency. Use the *adjacency-name* argument to define the name of the service.
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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(config-sbc-sbe-adj-sip)# fast-register disable	Disables fast register support on the SIP adjacency.
Step 21	**signaling-peer** *peer_address* **Example:** host1/Admin(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:** host1/Admin(config-sbc-sbe-adj-sip)# signaling-peer-port 5060	Specifies the remote signaling-peer port for the SIP adjacency to use.

	Command or Action	Purpose
Step 23	**account** *account_name* **Example:** host1/Admin(config-sbc-sbe-adj-sip)# account COREvlan	Defines the SIP adjacency as belonging to an account on an SBE.
Step 24	**registration target address** *host_address* **Example:** host1/Admin(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:** host1/Admin(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:** host1/Admin(config-sbc-sbe-adj-sip)# attach	Attaches the adjacency.
Step 27	**exit** **Example:** host1/Admin(config-sbc-sbe-adj-sip)# exit	Exits adj-sip mode to sbe mode.
Step 28	**show** **Example:** host1/Admin(config-sbc-sbe)# 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**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **group** *adjacency-group-name*
6. **exit**
7. **show**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`host1/Admin# configure`
		Enables global configuration mode.

Step 2	`sbc service-name`
	
	Example:
	`host1/Admin(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:
	`host1/Admin(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	`adjacency sip adjacency-name`
	
	Example:
	`host1/Admin(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:
	`host1/Admin(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	`exit`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-sip)# exit`
		Exits adj-sip mode to sbe mode.

Step 7	`show`
	
	Example:
	`host1/Admin(config-sbc-sbe)# show`
		Shows contents of configuration.

SIP Statistics Per Adjacency

The SIP Statistics Per Adjacency feature allows you to configure the collection of SIP message statistics at the level of adjacencies. Previously, no statistics were recorded for SIP adjacencies. With this feature enabled, you can tune the level of statistics setting on an SIP adjacency to get summary or detailed SIP statistics for each SIP adjacency.

Restrictions for SIP Statistics Per Adjacency

Some SIP messages are not assigned to an adjacency and, therefore, are not counted by the per-adjacency method statistics. This includes inbound requests that are rejected or dropped before being assigned to an adjacency.

Inbound SIP requests that are rejected include:

- Failure to identify essential headers or parameters in the request including:
    - Request-URI
    - Request method
    - Via header
    - CSeq header
    - To header
    - From header
    - Call-ID header
    - Contact header
- Failure to parse any header present in request, including:
    - All the headers listed under the previous bullet
    - Supported
    - Require
    - Content-Type
    - Replaces
    - Referred-By
    - RAck
    - Session-Expires
    - Event
- To: tag present but doesn't match SBC-specific format
- Route header present but cannot be parsed
- Transport parameter in Via header does not match received transport
- Resource failures in SIP stack
- Failure to match an in-dialog request to an existing dialog / transaction / subscription
- Glare detection

Inbound SIP requests that are dropped include:

- ACK to INVITE for which SBC has not transmitted a final response
- ACK to negative final response
- ACK not matched to INVITE transaction
- Retransmitted requests
- Pathological syntax errors (failure to identify message as a request)

The following inbound responses are dropped or converted to internal 5xx before being assigned to an adjacency:

> **Note** 2xx responses containing syntax errors are passed up to the SIP signaling stack and 3-6xx responses containing syntax errors are converted to internal 5xx responses before being passed up to the SIP signaling stack.

- Resource failures in SIP stack encountered when processing response
- 100 Trying responses
- CANCEL responses
- Retransmitted responses
- Syntax errors encountered when parsing / validating 1xx responses, including
  - Detection of multiple Via headers
  - Parse errors in any of the following headers, if present
    Require
    Supported
    Session-Expires
    Min-SE
    Expires
  - Pathological syntax errors (failure to identify message as a response)

Additionally, the following are not included in the statistics:

- Fast-register requests/response
- SIP PING requests/response

Configuring SIP Statistics Per Adjacency

Use the procedure in this section to configure the SIP Statistics per adjacency.

SUMMARY STEPS

1. **configure**
2. **sbc** *service-name*
3. **sbe**
4. **adjacency sip** *adjacency-name*
5. **statistics-setting** *summary/detail*

6. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`configure`
	
	Example:
	`host1/Admin# configure`
		Enables global configuration mode.

Step 2	`sbc` *service-name* **Example:** `host1/Admin(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:
	`host1/Admin(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	`adjacency sip` *adjacency-name* **Example:** `host1/Admin(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	`statistics-setting` *summary/detail* **Example:** `host1/Admin(config-sbc-sbe-adj-sip)# statistics-setting summary`	Enables summary statistics or detailed response-codes statistics recording on this SIP adjacency.

Step 6	`exit`
	
	Example:
	`host1/Admin(config-sbc-sbe-adj-sip)# exit`
		Exits adj-sip mode to sbe mode.

The following example shows the output of the **show services sbc sbe sip-method-stats** command.

```
host1/Admin# show services sbc one sbe sip-method-stats sip-inbound1 invite current15
SBC Service ''one''
Adjacency sip-inbound1 (SIP)
Statistics for SIP method INVITE
Total request recieved :0
Total request sent :0
Total 1xx response received :0
Total 1xx response sent :0
Total 2xx response received :0
Total 2xx response sent :0
Total 3xx response received :0
Total 3xx response sent :0
```

```
Total 4xx response received :0

Total 4xx response sent :0

Total 5xx response received :0

Total 5xx response sent :0

Total 6xx response received :0

Total 6xx response sent :0

Other response received :0

Other response sent :0
```

Configuration Examples for Implementing Adjacencies

This section provides the following configuration examples:

- Configuring an H.323 Adjacency: Examples
- Configuring a SIP Adjacency: Example

Configuring an H.323 Adjacency: Examples

This section contains two example H.323 adjacency configurations:

- H.323 Adjacency Example 1 (Two Gateways/Endpoints)
- H.323 Adjacency Example 2 (Gatekeeper in Network)

H.323 Adjacency Example 1 (Two Gateways/Endpoints)

In this example, the configuration is performed to support SBC peering with two H.323 gateways/endpoints in two different networks (as shown in shown in Figure 11-7).

Figure 11-7   SBC with Two H.323 Gateways/Endpoints in Two Different Networks
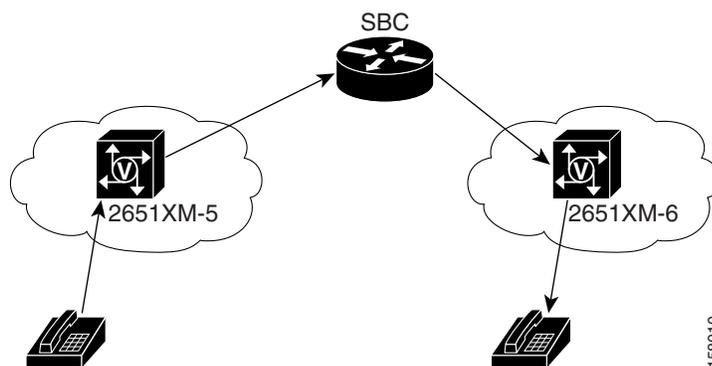
1. Go SBE submode.

    ```
    configure
     sbc umsbc-node10
     sbe
    ```

2. Configure an H.323 adjacency name and type (as a result, you will be in the H.323 adjacency submode).

```
adjacency h323 2651XM-5
```

3. Configure the local signaling address and port. All H.323 signaling traffic should be sent to this address and port by the gateway (default port is 1720).

```
signaling-address ipv4 88.88.137.10
signaling-port 5000
```

4. Configure the signaling address and port for the peer gateway. SBC will send all H.323 signaling traffic to this address and port.

```
signaling-peer 200.200.200.41
signaling-peer-port 5000
```

5. Restrict the set of remote signaling peers contacted over the adjacency. To be successful, the address of the endpoint originating or terminating the call should be within this subnet.

```
remote-address ipv4 200.200.200.0/24
```

6. Configure the account that the adjacency will belong to.

```
account core-side
```

7. Attach the H.323 adjacency. On attaching, the adjacency is activated.

(When attached, no additional configuration changes can be made to the adjacency. To make changes, execute the **no attach** command; then make changes and attach afterwards.)

```
attach
exit
```

8. Similarly, configure a second adjacency pointing to the gateway in another account, where the calls will terminate (or vice versa).

```
adjacency h323 2651XM-6
 signaling-address ipv4 88.88.137.10
 signaling-port 5050
 signaling-peer 10.10.119.12
 signaling-peer-port 5050
 remote-address ipv4 10.10.119.0/24
 account node-side
 attach
 exit
```

9. Use the **show** command to verify that the adjacency is attached.

H.323 Adjacency Example 2 (Gatekeeper in Network)

In this example, there is a gatekeeper (as shown in Figure 11-8) in the network. Therefore, everything is pointing to the gatekeeper on the remote side instead of the gateway; configuring a signaling port is not required.

The keyword **gk** is added to the **signaling-peer** command and an alias is configured. The rest of the configuration is the same as shown in "H.323 Adjacency Example 1 (Two Gateways/Endpoints)".

Figure 11-8   SBC with Two H.323 Gatekeepers in Two Different Networks
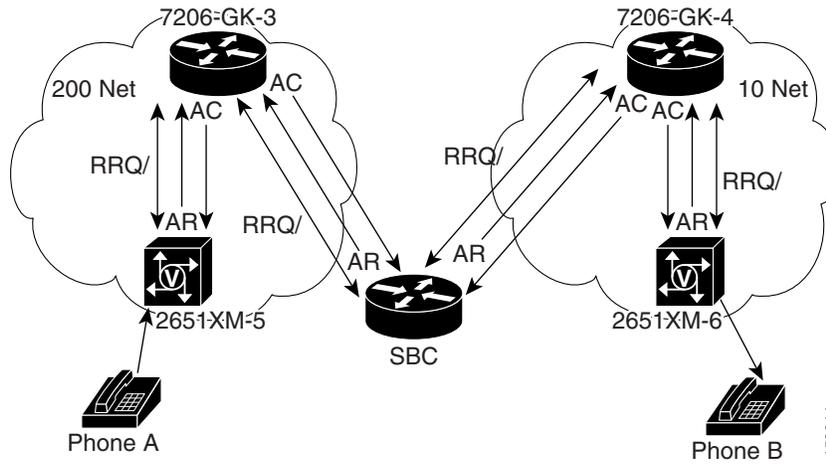
1. Go to SBE submode.

```
configure
sbc umsbc-node10
 sbe
```

2. Configure an H.323 adjacency name signaling-peer gatekeeper, and alias.

```
adjacency h323 GK-3
 signaling-peer gk 200.200.200.40
 alias SBC-GK3
```

3. Configure the local signaling address and port. All H.323 signaling traffic will be sent to this address.

```
signaling-address ipv4 88.88.137.10
signaling-port 5001
```

4. Restrict the set of remote signaling peers contacted over the adjacency. To be successful, the address of the endpoint originating or terminating the call should be within this subnet.

```
remote-address ipv4 200.200.200.0/24
```

5. Configure the account that the adjacency will belong to.

```
account core-side
```

6. Attach the H.323 adjacency. On attaching, the adjacency is activated.

   (When attached, no additional configuration changes can be made to the adjacency. To make changes, execute the **no attach** command; then make changes and attach afterwards.)

```
attach
exit
```

7. Similarly, configure a second adjacency pointing to the gateway in another account (as shown in Figure 11-8), where the calls will terminate (or vice versa).

```
adjacency h323 GK-4
alias SBC-GK4
 signaling-address ipv4 88.88.137.10
 signaling-port 5051
 signaling-peer gk 10.10.119.8
 remote-address ipv4 10.10.119.0/24
 account node-side
 attach
 exit
```

8. Use the **show** command to verify that the adjacency is attached.

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. Go to SBE submode:

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

2. Activate DBE, as follows:

    ```
    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
    ```

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.

SIP UAS Failure Detection enables the SBC 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* ping failures occur, then the device is considered to be unreachable. The signaling peer is considered offline as far as routing is concerned. The SBC sends pings at the rate specified in the period.

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

SUMMARY STEPS

1. **configure**
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`
	
	Example:
	`host1/Admin# configure`
		Enables global configuration mode.

Step 2	`sbc` *service-name* **Example:** `host1/Admin(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:
	`host1/Admin(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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(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:** host1/Admin(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) the SBC 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 SBC.

- Local removal of configuration in the SBC — If a SIP adjacency is deconfigured 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**

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`
	
	Example:
	`host1/Admin# configure`
		Enables global configuration mode.

Step 2	`sbc` *service-name* **Example:** `host1/Admin(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:
	`host1/Admin(config-sbc)# sbe`
		Enters the mode of an SBE entity within an SBC service.

Step 4	`adjacency sip` *adjacency-name* **Example:** `host1/Admin(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:** host1/Admin(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:** host1/Admin(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
```