The Accounting Server Connectivity Failure and Recovery Detection feature provides the scriptable option to reject new calls entering the VoIP network and tear down all existing calls upon detecting connectivity failure to the method list that is associated with the RADIUS-based accounting server(s).

Guide Contents
Troubleshooting Cisco IOS Voice Overview
Debug Command Output on Cisco IOS Voice Gateways
Filtering Troubleshooting Output
Cisco VoIP Internal Error Codes
Troubleshooting Cisco IOS Voice Telephony
Troubleshooting Cisco IOS Voice Protocols
Troubleshooting Cisco IOS Telephony Applications
Monitoring the Cisco IOS Voice Network
Cause Codes and Debug Values

Contents

- 1 Prerequisites for Accounting Server Connectivity Failure and Recovery Detection
- 2 Restrictions for Accounting Server Connectivity Failure and Recovery Detection
- 3 Global Accounting Script
- 4 How to Configure Accounting Server Connectivity Failure and Recovery Detection
  - 4.1 Configuring the GAS
    - 4.1.1 How the GAS Application Verifies Configuration Parameters
    - 4.1.2 SUMMARY STEPS
    - 4.1.3 DETAILED STEPS
  - 4.2 Loading the GAS
    - 4.2.1 Prerequisites
    - 4.2.2 SUMMARY STEPS
    - 4.2.3 DETAILED STEPS
  - 4.3 Starting the GAS
    - 4.3.1 Starting the GAS in Privileged EXEC Mode
    - 4.3.2 SUMMARY STEPS
    - 4.3.3 DETAILED STEPS
    - 4.3.4 Starting the GAS in Global Configuration Mode
    - 4.3.5 SUMMARY STEPS
    - 4.3.6 DETAILED STEPS
  - 4.4 Verifying the GAS
    - 4.4.1 SUMMARY STEPS
    - 4.4.2 DETAILED STEPS
  - 4.5 Troubleshooting Accounting Server Connectivity Failure and Recovery Detection
    - 4.5.1 SUMMARY STEPS
    - 4.5.2 DETAILED STEPS
- 5 Configuration Examples for Accounting Server Connectivity Failure and Recovery Detection
  - 5.1 Configuring the GAS: Example
    - 5.1.1 Figure: Example Topology
  - 5.2 Loading the GAS: Example
  - 5.3 Starting the GAS: Example
  - 5.4 Verifying the GAS: Example
    - 5.4.1 Sample Output from the show running-config Command
      - 5.4.1.1 Sample Output from the show voice accounting method Command

Prerequisites for Accounting Server Connectivity Failure and Recovery Detection

The following tasks are prerequisites to configuring the features:

- Establish a working IP network. For more information about configuring IP, refer to the Cisco IOS IP Address Services Configuration Guide.
- Configure VoIP. For more information about configuring VoIP, refer to the Voice Configuration Library.
- Configure a TFTP sever to perform storage and retrieval of the audio files, which are required by the Debit Card gateway or other features requiring Tool Command Language Interactive Voice Response (Tcl IVR) scripts and audio files.
- Program and configure the interface between the RADIUS server and the Cisco voice gateway to operate with VSAs.
- Create the accounting method list default that includes all RADIUS servers using the **aaa accounting connection default start-stop group radius** command in global configuration mode. This method list is required by the probe accounting records the Accounting Server Connectivity Failure and Recovery Detection feature uses to determine the state of connectivity to the method list.

Restrictions for Accounting Server Connectivity Failure and Recovery Detection

The Accounting Server Connectivity Failure and Recovery Detection feature is applicable only to the RADIUS accounting protocol. It is not applicable to any other protocols or servers, such as RADIUS access protocol, TACACS, or DIAMETER.

If both voice and dial calls need to be done on the same gateway, different accounting servers must be configured for each type of call.

Global Accounting Script

The Accounting Server Connectivity Failure and Recovery Detection feature uses a configurable, legless Global Accounting Script (GAS) to control algorithms that determine the state of connectivity to the method list that is associated with the accounting server. The Accounting Server Connectivity Failure and Recovery Detection feature has two major functional components:

- Tcl legless GAS-Controls and drives the detection, recovery, and probe algorithms.
- Application Tcl scripts-Performs the call treatments to incoming calls and existing calls when notified that the method list is unreachable.

The GAS is a Tcl script with configurable parameters that users can customize for their own network requirements. Users can configure one GAS for each method list, or one GAS script for multiple method lists.

Because the RADIUS accounting protocol is User Datagram Protocol (UDP)-based, it is connectionless, and there is no guaranteed connectivity with the RADIUS server. The Accounting Server Connectivity Failure and Recovery Detection feature uses the acknowledgment of accounting requests from the method list to detect connectivity.

The GAS determines the state transition of the method list and updates the AAA system with the latest method list status. If the method list is unreachable, the AAA system locates all the active calls associated

with the unreachable method list and informs the application script instances of the server unreachable event. The application script applies the appropriate treatments for this new event to the existing calls. For incoming calls, the application script checks the method list status and applies the appropriate treatment. For example, the application can clear the existing calls and reject new incoming calls for this method list. When the method list becomes reachable, the application script instances are notified, and they can take the appropriate action.

How to Configure Accounting Server Connectivity Failure and Recovery Detection

This section contains procedures identified as either required or optional.

- Configuring the GAS (optional)
- Loading the GAS (required)
- Starting the GAS (optional)
- Starting the GAS in Global Configuration Mode (optional)
- Verifying the GAS (optional)
- Troubleshooting Accounting Server Connectivity Failure and Recovery Detection (optional)

Configuring the GAS

Perform this task to configure the GAS.

How the GAS Application Verifies Configuration Parameters

The GAS application reads in the configured method list and uses it to identify the configuration parameters associated with that method list. Configuration parameters associated with a method list are either mandatory or optional.

If a mandatory configuration parameter does not exist, the GAS application displays the following message:

```
TCL GAS: >> Mandatory Parameter <parameter name from configuration avpair> does not exist
```

If a mandatory configuration parameter has an invalid type or value, the GAS application displays the following message:

```
TCL GAS:  >> Mandatory Parameter <exact parameter from configuration avpair> invalid value
```

At least one method list must be configured with the GAS application.

If the GAS application fails to read any of the mandatory configuration parameters, it fails and display this message:

```
TCL GAS:>>>> GasManager.Start  Exit Failure <<<<
```

SUMMARY STEPS

1. Download the GAS file.
2. Configure the GAS application and mandatory parameters.

DETAILED STEPS

Command or Action	Purpose	
1.	Download the GAS file.	Download the GAS file at Technical Support Software Download for Cisco Tool Command Language Software at http://www.cisco.com/cgi-bin/tablebuild.pl/tclware. The GAS file contains the GAS and a ReadMe file.
2.	Configure the GAS application and mandatory parameters.	Refer to the ReadMe file for script-specific information about configuration parameters, including which are mandatory and which are optional. • You must configure at least the mandatory parameters before loading the GAS. • You must configure the GAS for at least one method list. • For information on the changes to the Tcl application programming interface (API) included in the Accounting Server Connectivity Failure and Recovery Detection feature, refer to the TCL IVR Version 2.0 Programmer's Guide. **Note**: The GAS should be configured only by changing the value of its configuration parameters. The Technical Assistance Center (TAC) cannot support this feature, or the system on which it is loaded, if any changes are made to the script itself. Any changes to the GAS are supported by the Cisco Developer Support Program only, which requires a signed Developer Support Agreement.

Loading the GAS

Perform this task to load the GAS Tcl script.

Prerequisites

You must configure any script-specific parameters before loading the GAS.

SUMMARY STEPS

1. **enable**
2. **call application voice load** application-name

DETAILED STEPS

	Command or Action	Purpose
1.	**enable** Example: `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
2.	**call application voice load** application-name Example: `Router# call application voice load GAS`	Reloads the selected Tcl script from the URL.

Starting the GAS

You can start the GAS in privileged EXEC mode or global configuration mode. If it is started in privileged EXEC mode, it must be restarted every time the router is rebooted. If it is started in global configuration mode, it is saved in NVRAM and is started automatically when the router is rebooted.

Choose one of the following optional tasks to start the GAS.

- Starting the GAS in Privileged EXEC Mode (optional)
- Starting the GAS in Global Configuration Mode (optional)

Starting the GAS in Privileged EXEC Mode

Perform this task to start the GAS in privileged EXEC mode:

SUMMARY STEPS

1. **enable**
2. **call application session start** instance-name application-name

DETAILED STEPS

	Command or Action	Purpose
1.	**enable** Example: `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
2.	**call application session start** instance-name application-name Example: `Router# call application session start session_1 GAS`	Starts an instance of an application.

Starting the GAS in Global Configuration Mode

Perform this task to start the GAS in global configuration mode:

SUMMARY STEPS

1. **enable**
2. **configure** terminal
3. **call application session start** instance-name application-name

DETAILED STEPS

	Command or Action	Purpose
1.	**enable** Example: `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

2.	**configure** terminal Example: `Router# configure terminal`	Enters global configuration mode.
3.	**call application session start** instance-name application-name Example: `Router(config)# call application session start session_1 GAS`	Starts a new instance (session) of a Tcl IVR 2.0 application.

Verifying the GAS

Perform this task to verify that the GAS has been configured and is working correctly.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **show voice accounting method**

DETAILED STEPS

	Command or Action	Purpose
1.	**enable** Example: `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
2.	**show running-config** Example: `Router# show running-config`	(Optional) Displays the contents of the currently running configuration file. • Use the **show running-config** command to verify that the GAS parameters have the correct values.
3.	**show voice accounting method** [method-list-name] Example: `Router# show voice accounting method`	(Optional) Displays connectivity status information for a specified accounting method list or all the accounting method lists. • Use the method-list-name argument to specify a single method list, or omit this argument to display information for all method lists.

Troubleshooting Accounting Server Connectivity Failure and Recovery Detection

The Accounting Server Connectivity Failure and Recovery Detection feature uses a privileged EXEC mode command to enable diagnostic output concerning various events relating to gateway accounting Application Subscribe/Notify Layer (ASNL) to be displayed on a console. The **debug voice aaa asnl** command is intended only for troubleshooting purposes, because the volume of output generated by the software can result in severe performance degradation on the router.

The following procedure minimizes the load on the router created by the **debug voice aaa asnl** commands, because the console port is no longer generating character-by-character processor interrupts. If you cannot connect to a console directly, you can run this procedure via a terminal server. If you must break the Telnet

connection, however, you may not be able to reconnect because the router may be unable to respond due to the processor load of generating the **debug voice aaa asnl** output.

Perform the following task to minimize the impact of using the '*debug voice aaa asnl'* command.

SUMMARY STEPS

1. Attach a console directly to a router running Cisco IOS Release 12.3(8)T or a later release.
2. **enable**
3. **configure terminal**
4. **no logging console**
5. Use Telnet to access a router port and repeat Steps 2 and 3.
6. **terminal monitor**
7. **end**
8. **debug voice aaa asnl**
9. **configure** terminal
10. **no terminal monitor**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
1.	Attach a console directly to a router running Cisco IOS Release 12.2(8)T or a later release.	-
2.	**enable** Example: `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
3.	**configure terminal** Example: `Router# configure terminal`	Enters global configuration mode.
4.	**no logging console** Example: `Router(config)# no logging console`	Disables all logging to the console terminal. • To reenable logging to the console, use the **logging console** command in global configuration mode.
5.	Use Telnet to access a router port and repeat Steps 2 and 3, then continue with step 6.	Enters global configuration mode in a recursive Telnet session, which allows the output to be redirected away from the console port.
6.	**terminal monitor** Example: `Router(config)# terminal monitor`	Enables logging output on the virtual terminal.
7.	**end** Example:	Exits to privileged EXEC mode.

	`Router(config)# end`	
8.	**debug voice aaa asnl** Example: `Router# debug voice aaa asnl`	Displays debugging messages for gateway AAA ASNL. • Enter the **no debug voice aaa asnl** command when you are finished.
9.	'*configure* '**terminal** Example: `Router# configure terminal`	Enters global configuration mode.
10.	**no terminal monitor** Example: `Router(config)# no terminal monitor`	Disables logging on the virtual terminal.
11.	**end** Example: `Router(config)# end`	Exits to privileged EXEC mode.

Configuration Examples for Accounting Server Connectivity Failure and Recovery Detection

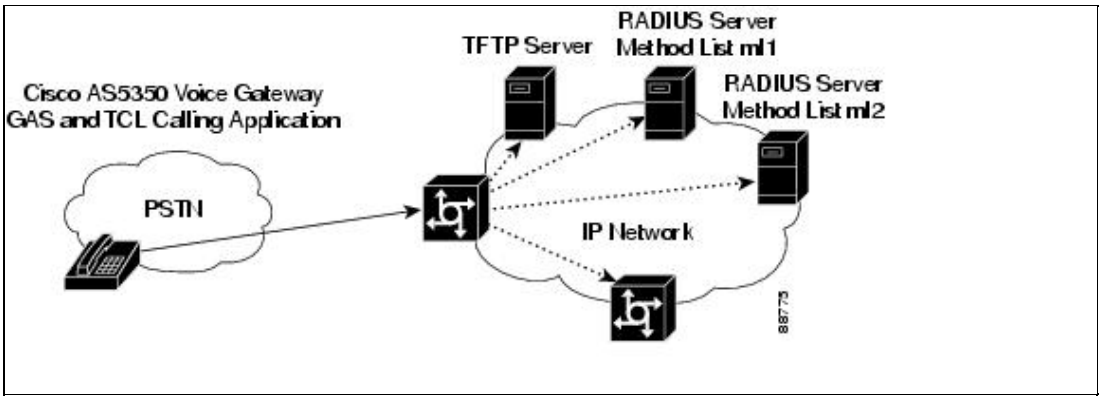
This section provides configuration examples to match the identified configuration tasks in the previous section:

- Configuring the GAS: Example
- Loading the GAS: Example
- Starting the GAS: Example
- Verifying the GAS: Example

Configuring the GAS: Example

Figure: Example Topology shows the topology used in this example.

Figure: Example Topology



In this example, the GAS is configured for two method lists: ml1 and ml2.

```
call application voice GAS tftp://192.255.254.253/app_GAS.2.0.0.0.tcl
call application voice GAS method-list ml1;ml2
call application voice GAS gas-active-timer-ml1 30
call application voice GAS detect-failure-responses-ml1 2
call application voice GAS recovery-responses-ml1 2
call application voice GAS probe-retry-timer-ml1 5
call application voice GAS report-accounting-failed-ml1 false
call application voice GAS probe-user-name-ml1 johndoe
call application voice GAS acct-inactivity-period-ml1 120
call application voice GAS send-accounting-on-ml1 true
call application voice GAS use-gas-debugs-ml1 true
call application voice GAS gas-active-timer-ml2 30
call application voice GAS detect-failure-responses-ml2 2
call application voice GAS recovery-responses-ml2 5
call application voice GAS probe-retry-timer-ml2 10
call application voice GAS report-accounting-failed-ml2 true
call application voice GAS acct-inactivity-period-ml2 90
call application voice GAS send-accounting-on-ml2 false
call application voice GAS use-gas-debugs-ml2 false
```

The ml1 method list is configured with the following parameter values:

Parameter Value	Description
gas-active-timer = 30	Generate a syslog message indicating that the GAS application is active every 30 minutes.
detect-failure-responses-ml1 = 2	After receiving two consecutive failed responses, the method list is declared unreachable.
recovery-responses-ml1 = 2	After receiving two consecutive success responses, the method list is declared reachable again.
probe-retry-timer-ml1 = 5	When the method list is in an unreachable state, send probe accounting records every 5 seconds.
report-accounting-failed-ml1 = false	The application script should not be notified when accounting fails before the method list is marked as unreachable.
probe-user-name-ml1 = johndoe	Use johndoe as the user name field in the probe accounting records.
acct-inactivity-period-ml1 = 120	When the method list is in a reachable state and 120 seconds of inactivity have passed after the last accounting record was sent to the method list, a probe accounting record is sent to determine connectivity to the method list.
send-accounting-on-ml1 = true	A message is sent when the method list transitions from unreachable to reachable. This behavior should be synchronized with the calling application script. If the application script tears down the existing calls when it is notified that the method list status transitioned from reachable to unreachable, this parameter should be set to true; otherwise, it should be set to false.
use-gas-debugs-ml1 = true	Debug output is controlled by the **debug voip ivr** commands and displayed to the terminal. If the use-gas-debugs parameter was set to false, only the value of the use-gas-debugs parameters for the ml1 and ml2 method lists would be displayed. The default value is true.

Loading the GAS: Example

In the following example, the GAS named GAS is loaded:

```
enable
```

```
call application voice load GAS
```

Starting the GAS: Example

In the following example, the GAS named GAS is started in instance session_1:

```
enable
call application session start session_1 GAS
```

Verifying the GAS: Example

In the following examples, the output is displayed for each command in the task.

Sample Output from the show running-config Command

In the following example, method lists ml1 and ml2 are defined, and the GAS named GAS is configured:

```
Router# show running-config
Current configuration :4419 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname as5300-2
!
logging buffered warnings
enable password cisco
!
resource-pool disable
clock timezone gmt 15 40
!
aaa new-model
!
aaa group server radius ml1
 server 10.8.159.105 auth-port 1645 acct-port 1646
aaa group server radius ml2
 server 10.9.57.101 auth-port 1715 acct-port 1716
!
aaa accounting update newinfo
aaa accounting connection default start-stop group radius
aaa accounting connection h323 start-stop group radius
aaa accounting connection ml1 start-stop group ml1
aaa accounting connection ml2 start-stop group ml2
aaa session-id common
ip subnet-zero
!
isdn switch-type primary-5ess
!
mta receive maximum-recipients 0
no memory check-interval
!
!
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
```

```
!
controller T1 1
 framing esf
 clock source line secondary 1
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 2
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 3
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
gw-accounting aaa
 method ml1
!
interface Loopback1
 no ip address
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet0
 ip address 10.8.156.2 255.255.0.0
!
!
interface Serial0:23
 no ip address
 dialer-group 1
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface Serial1:23
 no ip address
 isdn switch-type primary-5ess
 no cdp enable
!
interface Serial2:23
 no ip address
 isdn switch-type primary-5ess
 no cdp enable
!
interface Serial3:23
 no ip address
 isdn switch-type primary-5ess
 no cdp enable
!
interface FastEthernet0
 ip address 172.19.141.84 255.255.0.0
 ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
!
ip default-gateway 10.8.0.1
ip classless
ip route 10.7.0.0 255.255.0.0 10.8.0.1
ip route 10.8.0.1 255.255.255.255 Ethernet0
ip route 10.9.0.0 255.255.0.0 10.8.0.1
```

```
ip route 192.255.254.253 255.255.255.255 10.8.0.1
ip route 192.255.254.254 255.255.255.255 10.8.0.1
no ip http server
!
access-list 101 permit ip any any
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
!
radius-server host 10.8.159.105 auth-port 1645 acct-port 1646
radius-server host 10.9.57.101 auth-port 1715 acct-port 1716
radius-server key cisco
radius-server authorization permit missing Service-Type
radius-server vsa send accounting
radius-server vsa send authentication
call rsvp-sync
!
call application voice GAS tftp://192.255.254.253/app_GAS.2.0.0.0.tcl
call application voice GAS method-list ml1;ml2
call application voice GAS gas-active-timer-ml1 5
call application voice GAS detect-failure-responses-ml1 2
call application voice GAS recovery-responses-ml1 2
call application voice GAS probe-retry-timer-ml1 5
call application voice GAS report-accounting-failed-ml1 false
call application voice GAS probe-user-name-ml1 johndoe
call application voice GAS acct-inactivity-period-ml1 120
call application voice GAS send-accounting-on-ml1 true
call application voice GAS use-gas-debugs-ml1 true
!
call application voice GAS gas-active-timer-ml2 5
call application voice GAS detect-failure-responses-ml2 2
call application voice GAS recovery-responses-ml2 5
call application voice GAS probe-retry-timer-ml2 10
call application voice GAS report-accounting-failed-ml2 true
call application voice GAS acct-inactivity-period-ml2 90
call application voice GAS send-accounting-on-ml2 false
call application voice GAS use-gas-debugs-ml2 false
!
call application voice calling_app tftp://192.255.254.253/app_session_rw.tcl
!
call application session start G1 GAS
!
voice-port 0:D
!
voice-port 1:D
!
voice-port 2:D
!
voice-port 3:D
!
!
mgcp profile default
!
dial-peer cor custom
!
!
dial-peer voice 1 pots
 application calling_app
 incoming called-number 25170
 port 0:D
!
dial-peer voice 1800877 voip
 destination-pattern 1800877....
 session target ipv4:10.8.156.3
!
```

```
alias exec osperr debug voip sett err
alias exec h225 debug cch323 h225
alias exec h245 debyg cch323 h245
alias exec ivr debyg voip ivr
alias exec ccapi debub voip ccapi in
!
line con 0
 exec-timeout 0 0
 logging synchronous
line aux 0
line vty 0
 password lab
line vty 1 4
!
end
```

Sample Output from the show voice accounting method Command

The following example displays the status history for the ml1 method list:

```
Router# show voice accounting method
Accounting Method List [ml1]
=====================
Current Status:
---------------
unreachable                [21:52:39 gmt Dec 4 2002]
last record sent time      [23:14:59 gmt Dec 4 2002]
total probe sent out       [84]
Status History:
---------------
(2) unreachable            [21:52:39 gmt Dec 4 2002]
(1) reachable              [21:46:19 gmt Dec 4 2002]
                SUCCESS                          FAILURE
Record  [Received   | Notified ] [Received   |  Notified  | Reported ]
Type    [from server| to client] [from server|  to client | to call  ]
------  [-----------|----------] [-----------|------------|----------]
START   [     0     |     0    ] [     0     |      0     |     0    ]
UPDATE  [     0     |     0    ] [     0     |      0     |     0    ]
STOP    [     0     |     0    ] [     84    |      84    |     0    ]
ACCT_ON [     0     |     0    ] [     0     |      0     |     0    ]
------  [-----------|----------] [-----------|------------|----------]
TOTAL   [     0     |     0    ] [     84    |      84    |     0    ]
```